Numerical analysis and development of accurate models in a CFD solver dedicated to naval applications with waves

Théo Descamps

THÈSE DE DOCTORAT DE

L'ÉCOLE CENTRALE DE NANTES

ÉCOLE DOCTORALE N° 602
Sciences pour l'Ingénieur
Spécialité : *Mécanique des Milieux Fluides*

Par
Théo DESCAMPS

Numerical analysis and development of accurate models in a CFD solver dedicated to naval applications with waves

Thèse présentée et soutenue à Nantes, le 15 novembre 2022
Unité de recherche : UMR 6598, Laboratoire de recherche en Hydrodynamique, Énergétique et Environnement Atmosphérique (LHEEA)

Rapporteurs avant soutenance :

Bettar Ould EL MOCTAR	Full professor, University of Duisburg-Essen (Germany)
Kevin J. MAKI	Associate Professor, University of Michigan (USA)

Composition du Jury :

Président :	Antonio SOUTO-IGLESIAS	Full Professor, Universidad Politécnica de Madrid (Spain)
Examinateurs :	Andrea DI MASCIO	Full Professor, Università degli Studi dell'Aquila (Italy)
	Giuseppina COLICCHIO	Senior researcher, National Research Council (Italy)
Dir. de thèse :	David LE TOUZÉ	Professeur des universités, École Centrale de Nantes
Co-encadrant :	Benjamin BOUSCASSE	Chargé de recherche, École Centrale de Nantes
Co-encadrant :	Lionel GENTAZ	Maître de Conférences, École Centrale de Nantes

ACKNOWLEDGEMENT

First, I want to thank my supervisors David, Lionel, and Benjamin for accompanying me throughout my Ph.D. I want to give particular thanks to Ben for always believing in me even when I doubted myself.

Then, I thank Pierrick and all the BVS-M&O team for helping me with this tricky thing that is the CFD mesh, for giving me confidence in my abilities, and for the good times spent together.

I naturally give a lot of thanks to Zhaobin, Young-Myung, and Young-Jun for enlightening me on the dark and winding road to OpenFOAM simulations. Thank you also to Sithik that have shared this road with me.

It would be hard not to thank my friends Maxime and Gaspard with who we have spent time sharing our pains as well as our successes in this long trip that is the Ph.D. thesis. I also thank Gaël with who we spent time far from the Ph.D. coasts closer to the Sea Shanty world.

More generally, I would like to thank the LHEEA members for their kindness and the warm atmosphere in the laboratory. And thank you to the BV-M&O team for the time spent working together with the research Chair ECN-BV.

Finally, I would thank Professor El Moctar and Professor Maki for accepting to be reviewers of my thesis. And I express my gratitude to Professor Souto-Iglesias, Professor Di Mascio, and Doctor Colicchio for taking part in my examination panel.

Contents

1 Introduction **1**

1.1 Previous and related researches 2

1.1.1 Choice of mathematical models 2

1.1.2 Choice of the numerical solver 3

1.1.3 Evaluation of the performances of a solver 5

1.2 Present Contributions 6

1.3 Thesis outline 7

I Theoretical and numerical modeling **9**

2 Mathematical models **10**

2.1 Two phase incompressible flow 10

2.1.1 Single-phase Navier-Stokes equations 10

2.1.2 Two-phase single-field Navier-Stokes equations 11

2.1.3 Two phase single-field Reynolds Average Navier-Stokes equations . . . 13

2.1.4 VOF-RANSE equations with a moving referential 15

2.2 Fluid-structure interactions 15

2.2.1 Wall boundary 15

2.2.2 Rigid body motions 16

2.3 Wave generation 19

2.3.1 Wave model 19

2.4 The Spectral Wave Explicit Navier-Stokes Equations (SWENSE) 23

3 Numerical methods **27**

3.1 Finite Volume spatial discretisation 27

3.1.1 Mesh characteristics 27

3.1.2 Cell and face center values 29

3.1.3 Cell to Face center interpolation . . . 29
3.1.4 Face to Cell center reconstruction . . . 34
3.1.5 Divergence . . . 34
3.1.6 Surface normal gradient . . . 35
3.1.7 Non-orthogonal correction . . . 37
3.1.8 Cell center gradient . . . 37
3.2 Time discretisation . . . 38
3.2.1 Implicit Euler and second-order Backward schemes . . . 39
3.2.2 Crank-Nicolson schemes . . . 40
3.2.3 Finite-volume method and moving meshes . . . 42
3.3 Solver algorithm . . . 43
3.3.1 Step one of PIMPLE loop: Body and mesh motion calculation . . . 48
3.3.2 Step two of PIMPLE loop: VOF resolution . . . 57
3.3.3 Discussion on VOF resolution . . . 63
3.3.4 PISO loop: RANSE resolution . . . 65
3.3.5 Turbulence model resolution . . . 69
3.3.6 Boundary conditions . . . 70
3.3.7 Matrix equation resolution . . . 72
3.3.8 Flow chart . . . 74
3.4 Iterative error . . . 77
3.4.1 Iterative error in V&V procedure . . . 77
3.5 Numerical solver modifications and new implementations . . . 79
3.5.1 Backward MULES . . . 79
3.6 Reference numerical set-up . . . 84
3.6.1 Numerical schemes . . . 84
3.6.2 Numerical algorithms . . . 85

II foamStar numerical model accuracy 88

4 Taylor-Green vortices 89
4.1 Case definition . . . 89
4.2 Influence of the time schemes on the convergence . . . 90
4.3 Influence of mesh deformations on the spatial convergence with and without skewness corrector . . . 93

4.3.1 Mesh types . . . 93
4.3.2 Results . . . 102
4.4 Conclusion . . . 108

5 Periodic regular wave **109**
5.1 Error definition . . . 111
5.2 Numerical set-up . . . 111
5.3 Meshes . . . 112
5.4 Steep regular wave propagation accuracy with backward and Crank-Nicolson MULES . . . 117
5.4.1 Field convergence . . . 117
5.4.2 Error in amplitude within the tested time and space resolution range with Crank-Nicolson 1 and backward time schemes . . . 121
5.4.3 Error in amplitude within the tested time and space resolution range with Crank-Nicolson 0.95 and backward time schemes . . . 122
5.4.4 Spatial and temporal MULES . . . 128
5.4.5 Simulation with non-uniform mesh . . . 130
5.5 Regular wave propagation accuracy depending on the wave steepness and the cell aspect ratio . . . 132
5.5.1 Optimal free-surface aspect ratio . . . 132
5.5.2 Comparison of the errors with backward and Crank-Nicolson 0.95 schemes 135
5.6 Comparison of the efficiency of various discretizations . . . 137
5.7 Conclusion . . . 139

6 Wave propagation with relaxation zones **141**
6.1 Case definition . . . 141
6.2 Error definition . . . 144
6.2.1 Amplitude error . . . 144
6.2.2 Phase shift . . . 144
6.2.3 Wave period . . . 144
6.3 Numerical setup . . . 145
6.4 Meshes . . . 145
6.5 Simulation in fixed frame with Navier-Stokes model . . . 146
6.5.1 Time histories . . . 146
6.5.2 Generation of vortices in the air . . . 148

6.5.3 Optimal free-surface cell aspect ratio and order of convergence 150
6.6 Simulation in fixed frame with RANS model 153
6.6.1 Influence of the turbulent model on the air vortices 153
6.6.2 Removing the inlet relaxation zone 155
6.7 Simulation in moving frame with RANS model 156
6.8 Efficiency 160
6.9 Conclusion 162

III Naval application 166

7 Naval simulation in head waves 167
7.1 Case definition 167
7.1.1 Geometry and test conditions 167
7.1.2 Numerical Setup 169
7.2 First assessment with regular head waves 170
7.2.1 Settings 170
7.2.2 Meshes configuration 172
7.2.3 Regular head waves numerical results 176
7.3 Ship in long crested irregular waves 181
7.4 Application of the recommendations 183
7.4.1 Mesh configuration 183
7.4.2 Settings 186
7.4.3 Results 187
7.4.4 Overview of the simulation improvements 193
7.5 Conclusion 193

8 Application in the industrial context 195
8.1 Quick integration with applicable guidelines 195
8.1.1 Fields of application and recommendations 196
8.1.2 Simplified setup 196
8.1.3 User guides 198
8.2 Computational cost managment through the control of the residuals 198
8.2.1 Existing PIMPLE residual controllers 200
8.2.2 Proposition of a new PIMPLE controller 201

9 Conclusion **207**
9.1 Overview of the present work 207
9.1.1 Analysis of the numerical implementation 207
9.1.2 Accuracy simulating nonlinear regular waves 208
9.1.3 Naval applications 209
9.2 Perspectives and proposal for future works 210

Bibliography **210**

List of Figures

2.1 Representation of the roll, pitch and yaw (ϕ, θ, ψ) rotations of boat referential (C_g, $\varepsilon_{\mathbf{x}}$, $\varepsilon_{\mathbf{y}}$, $\varepsilon_{\mathbf{z}}$) regarding to Galilean referential (O, $\mathbf{e_x}$, $\mathbf{e_y}$, $\mathbf{e_z}$) 16
2.2 Relaxation-zone weight (w) distribution along a simulation domain 22
2.3 The SWENSE method decomposes the total field into an incident part and a complementary part 24

3.1 Two neighboring control volumes sharing a face f from an unstructured 3D mesh 28
3.2 Flux limiter Ξ as a function of r with TVD first-order and second-order regions 32
3.3 A cell face of a non-orthogonal mesh 36
3.4 *foamStar* simplified flowchart 47
3.5 Face decomposition used for the calculations of surface areas and geometric-center of faces. With F the geometric center and $\widetilde{\mathrm{F}}$ the barycenter of vertices. 53
3.6 Cell decomposition used for the calculations of volume and geometric-center of cells. With P the geometric center and $\widetilde{\mathrm{P}}$ the barycenter of face centers. 54
3.7 Face-swept volume decomposition used for mesh flux computation 55
3.8 Two sub-face-swept decomposition required for mesh flux computation 57
3.9 Simplified flowchart of the PISO loop 66
3.10 *foamStar* global flowchart 76

4.1 Temporal convergence of the velocity and pressure error for various temporal schemes at $t = 1$ s with `U256x256`. 92
4.2 `stretch` type mesh `U16X64`. 94
4.3 `twist` deformation $\theta_{\max} = \pi$ applied on a `U64X64` mesh. 96
4.4 Evolution of the maximal and average aspect ratios depending on the mesh resolution for various maximal angles of `twist` deformation. 97
4.5 Evolution of the maximal and average non-orthogonality angle depending on the mesh resolution for various maximal angles of `twist` deformation. 98

4.6 Evolution of the maximal and average mesh skewness depending on the mesh resolution for various maximal angles of `twist` deformation. 98
4.7 `zigZag` deformation applied on a `U32X32` mesh. 99
4.8 Evolution of the maximal and average aspect ratio depending on the mesh resolution for various coefficients of `zigZag` deformation. 100
4.9 Evolution of the maximal and average non-orthogonality angle depending on the mesh resolution for various coefficients of `zigZag` deformation. 101
4.10 Evolution of the maximal and average mesh shewness depending on the mesh resolution for various coefficients of `zigZag` deformation. 101
4.11 Spatial convergence of the velocity and pressure error for various aspect ratios *AR* using the `stretch` deformation. 103
4.12 Spatial convergence of the velocity and pressure error for various maximal angles of `twist` deformation. 104
4.13 Spatial convergence of the velocity and pressure error with and without OpenFOAM skewness corrector for various coefficients of `zigZag`. 106
4.14 Spatial convergence of the velocity and pressure intrinsic error with and without OpenFOAM skewness corrector for various coefficients of `zigZag`. 107

5.1 2D Periodic domain configuration for regular wave propagation. 110
5.2 Uniform and non-uniform mesh profiles used for periodic-wave propagation in periodic domain. 113
5.3 Non-uniform stretched mesh `NU65x256` with refinement zones identification. . 114
5.4 Surface elevation profile with backward, Crank-Nicolson 0.95 and Crank-Nicolson 1 at $t = 10T$; depending on the discretization with $\mathrm{Cfl_x} \approx 0.05$; periodic wave `H01`; uniform mesh. 118
5.5 Regular wave horizontal velocity field in periodic domain with uniform mesh `U256x256`; $\Delta t = T/800$. 119
5.6 Regular wave dynamic pressure field in periodic domain with uniform mesh `U256x256`; $\Delta t = T/800$. 119
5.7 Horizontal velocity profile with backward, Crank-Nicolson 0.95 and Crank-Nicolson 1 at $t = 10T$ and at the crest of the wave; depending on the discretization with $\mathrm{Cfl_x} \approx 0.05$; periodic wave `H01`; uniform mesh 120
5.8 *Iso-time-step view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number $\mathrm{Cfl_x}$ for various time steps Δt with backward and Crank-Nicolson; periodic wave `H01`; uniform mesh. 122

5.9 *Iso-time-step view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number Cfl_x for various time steps Δt with backward and Crank-Nicolson 0.95; periodic wave `H01`; uniform mesh. 123

5.10 *Iso-cell-size view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number Cfl_x for various cell sizes Δx with backward and Crank-Nicolson 0.95; periodic wave `H01`; uniform mesh. 123

5.11 Total kinetic and potential energy in the domain over time with backward and Crank-Nicolson 0.95 for several time steps; periodic wave `H01`; uniform mesh `U256x256`. 125

5.12 *Iso-time-step view* of the surface-elevation first-harmonic amplitude intrinsic error at $t = 37.5T$ depending on Courant number Cfl_x for various time steps Δt with backward and Crank-Nicolson 0.95; periodic wave `H01`; uniform mesh. 126

5.13 *Iso-cell-size view* of the surface-elevation first-harmonic amplitude intrinsic error at $t = 37.5T$ depending on Courant number Cfl_x for various cell sizes Δx with backward and Crank-Nicolson 0.95; periodic wave `H01`; uniform mesh. 127

5.14 Surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on cell discretization at fixed Courant number with backward and Crank-Nicolson 0.95; periodic wave `H01`; uniform mesh. 128

5.15 *Iso-time-step view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number Cfl_x for various time steps Δt with Crank-Nicolson 0.95 `MULES` and `S-MULES` algorithms; periodic wave `H01`; uniform mesh. 129

5.16 *Iso-time-step view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number Cfl_x for various time steps Δt with backward `MULES` and `S-MULES` algorithms; periodic wave `H01`; uniform mesh. 129

5.17 Regular wave horizontal velocity field in periodic domain with uniform and non-uniform meshes `U128x128` and `NU128x128`; backward `MULES`; $\Delta t = T/400$. 130

5.18 *Iso-time-step view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number Cfl_x for various time steps Δt for CrankNicolson 0.95 with Uniform mesh and non-uniform refined mesh; periodic wave `H01`; uniform mesh. 131

5.19 *Iso-time-step view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number $\mathrm{Cfl_x}$ for various time steps Δt for backward with uniform and non-uniform mesh; periodic wave `H01`; $\mathrm{AR_{FS}}$=1. 132

5.20 Surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on cell aspect ratio at free-surface ($\mathrm{AR_{FS}}$) at various Courant number $\mathrm{Cfl_x}$ for backward and Crank-Nicolson 0.95; periodic wave; non-uniform mesh. 134

5.21 *Iso-time-step view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number $\mathrm{Cfl_x}$ for various time steps Δt with backward and Crank-Nicolson 0.95; periodic wave `H01`; uniform mesh; $\mathrm{AR_{FS}}$=1. 135

5.22 *Iso-time-step view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number $\mathrm{Cfl_x}$ for various time steps Δt with backward and Crank-Nicolson 0.95; periodic wave `H005`; uniform mesh; $\mathrm{AR_{FS}}$=4. 136

5.23 *Iso-time-step view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number $\mathrm{Cfl_x}$ for various time steps Δt with backward and Crank-Nicolson 0.95; periodic wave `H001`; uniform mesh; $\mathrm{AR_{FS}}$=8. 136

5.24 *Iso-cell-size view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ and CPU time for various cell sizes Δx with backward and Crank-Nicolson 0.95; periodic wave `H01`; non-uniform mesh; $\mathrm{AR_{FS}}$=1. 137

5.25 *Iso-cell-size view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ and CPU time for various cell sizes Δx with backward and Crank-Nicolson 0.95; periodic wave `H005`; non-uniform mesh; $\mathrm{AR_{FS}}$=4. 138

5.26 *Iso-cell-size view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ and CPU time for various cell sizes Δx with backward and Crank-Nicolson 0.95; periodic wave `H001`; non-uniform mesh; $\mathrm{AR_{FS}}$=8. 138

5.27 *Iso-cell-size view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ and CPU time for various cell sizes Δx with $\mathrm{AR_{FS}}$=4 and $\mathrm{AR_{FS}}$=8; periodic wave `H001`; non-uniform mesh; backward. 139

6.1 2D domain dimensions used for pure wave propagation with Navier-Stokes model. (`RZi1o2`) 143

6.2 2D domain dimensions used for pure wave propagation with RANS model. (`RZo3`) 143

6.3 non-uniform stretched mesh `NU32x128` used for wave propagation with relaxation zones. 145

6.4 Time histories of the surface elevation first-harmonic amplitude error, phase shift and wave periods for various time steps Δt with backward and Crank-Nicolson 0.95; regular wave; `H01`; `NU256x256` and `NU128x128`; Navier-Stokes model; domain `RZi1o2`. 147

6.5 Regular wave horizontal velocity field with backward time scheme at $t = 10T$; `NU256x256`; $\Delta t = T/400$; Navier-Stokes model; domain `RZi1o2`. 149

6.6 Regular wave vorticity magnitude field with backward time scheme at $t = 10T$; `NU256x256`; $\Delta t = T/400$; Navier-Stokes model; domain `RZi1o2`. 149

6.7 Surface elevation first-harmonic amplitude error at $t = 37.5T$ depending on cell aspect ratio at the free surface (AR_{FS}) for two Courant number Cfl_x and for backward and Crank-Nicolson 0.95; Navier-Stokes model, domain `RZi1o2`. . . 152

6.8 Regular wave vorticity magnitude field with backward time scheme at $t = 10T$; `NU256x256`; $\Delta t = T/400$, RANSE $k-\omega$ SST; domain `RZi1o2`. 153

6.9 Regular wave volume-fraction field with backward time scheme at $t = 10T$; `NU256x256`; $\Delta t = T/400$, RANS k-ω-SST; domain `RZi1o2`. 154

6.10 Regular wave vorticity magnitude field with backward time scheme at $t = 10T$; `NU256x256`; $\Delta t = T/400$, RANS fs-k-ω-SST; domain `RZi1o2` 155

6.11 Regular wave vorticity magnitude field with backward time scheme at $t = 10T$; `NU256x256`; $\Delta t = T/400$, RANS fs-k-ω-SST; domain `RZi1o2` 155

6.12 Regular wave vorticity magnitude field with backward time scheme at $t = 10T$; `NU256x256`; $\Delta t = T/400$, RANS fs-k-ω-SST; domain `RZo3` (no inlet relaxation zone). 156

6.13 Regular wave vorticity magnitude field with backward time scheme at $t = 10T$; `NU256x256`; $\Delta t = T/400$, RANS fs-k-ω-SST, without inlet relaxation zone; domain `RZo3`. 156

6.14 Regular wave vorticity magnitude field with backward time scheme at $t = 20Te$; `NU256x256`; $\Delta t = T/400$, RANS fs-k-ω-SST; domain `RZo3`. 157

6.15 *Iso-time-step view* of the surface elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number Cfl_x for various time steps Δt with fixed and moving frame; RANS fs-k-ω-SST; `H01`; AR_{FS}=2; backward. 158

6.16 *Iso-time-step view* of the surface elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number Cfl_x for various time steps Δt with fixed and moving frame; RANS fs-k-ω-SST; `H005`; AR_{FS}=4; backward. 158

6.17 *Iso-time-step view* of the surface elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number $\mathrm{Cfl_x}$ for various time steps Δt with fixed and moving frame; RANS fs-k-ω-SST; `H001`; $\mathrm{AR_{FS}}$=4; backward. 159
6.18 *Iso-cell-size view* of the surface elevation first-harmonic amplitude error at $t = 37.5T$ and CPU time for various cell sizes Δx with fixed and moving frame; RANS fs-k-ω-SST; `H01`; $\mathrm{AR_{FS}}$=2; backward. 160
6.19 *Iso-cell-size view* of the surface elevation first-harmonic amplitude error at $t = 37.5T$ and CPU time for various cell sizes Δx with fixed and moving frame; RANS fs-k-ω-SST; `H005`; $\mathrm{AR_{FS}}$=4; backward. 161
6.20 *Iso-cell-size view* of the surface elevation first-harmonic amplitude error at $t = 37.5T$ and CPU time for various cell sizes Δx with fixed and moving frame; RANS fs-k-ω-SST; `H001`; $\mathrm{AR_{FS}}$=8; backward. 161

7.1 Numerical domain for KCS simulations 169
7.2 Relaxation zone configuration for KCS simulations 171
7.3 Side view of the mesh at the symmetry plane and on the hull; KCS simulation; The red box identifies the free-surface zone. Blue box is a zoom on the stern of the ship. 172
7.4 Top view of the mesh at $z = 0$ m; KCS simulation; colors identify some refinement zones. 173
7.5 Side views of the meshes used for simulations with calm water ; KCS; Fr=0.26 174
7.6 Wave elevation profile at $y = L_{PP}$ and at the symmetry plane ($y = 0$); KCS Fr = 0.26; regular head wave $\lambda = 0.65L_{PP}$; steepness $\varepsilon = 1.6\%$ 176
7.7 KCS heave and pitch 0^{th}, 1^{st} and 2^{nd} harmonic amplitude, Fr=0.26, head waves . 178
7.8 KCS total resistance and added resistance, Fr=0.26, head waves 180
7.9 Comparison of the KCS wake in head wave (C4) using *foamStar* and *foamStar-SWENSE* 181
7.10 KCS wake in irregular head wave (SS6), Fr=0.26 182
7.11 KCS motions (heave and pitch) and total resistance response in irregular head wave (SS6), Fr=0.26 183
7.12 Side view of the improved mesh at the symmetry plane and on the hull; KCS simulation; The red box identifies the free-surface zones. Blue box is a zoom on the stern of the ship. 185
7.13 Top view of the improved mesh at $z = 0$ m; KCS simulation; colors identify some refinement zones. 186
7.14 Relaxation zone configuration for KCS simulations without inlet forcing zone . 187

7.15 Wave elevation profile at $y = L_{PP}$ and at the symmetry plane ($y = 0$) using Euler, second-order backward and Crank-Nicolson 0.95 time schemes; KCS Fr = 0.26; regular head wave $\lambda = 0.65L_{PP}$; steepness $\varepsilon = 1.6\%$; Improved configuration. . 189
7.16 Comparison of the wave elevation map obtained using Euler and second-order backward time schemes at $t = 20T_e$; KCS Fr = 0.26; regular head wave $\lambda = 0.65L_{PP}$; steepness $\varepsilon = 1.6\%$ 190
7.17 Comparison of the wave elevation map obtained using Crank-Nicolson 0.95 and second-order backward time schemes at $t = 20T_e$; KCS Fr = 0.26; regular head wave $\lambda = 0.65L_{PP}$; steepness $\varepsilon = 1.6\%$. 190
7.18 Time history of the pressure resistance and viscous resistance coefficients using Euler, second-order backward and Crank-Nicolson 0.95; KCS Fr = 0.26; regular head wave $\lambda = 0.65L_{PP}$; steepness $\varepsilon = 1.6\%$. 191
7.19 Time history of the heave and pitch using Euler, second-order backward and Crank-Nicolson 0.95; KCS Fr = 0.26; regular head wave $\lambda = 0.65L_{PP}$; steepness $\varepsilon = 1.6\%$. 192

8.1 BVS M&O *foamStar* setup procedure 197
8.2 Initial residual of the pressure equation and linear solver iteration distribution along two time steps; KCS in head regular wave; Fr=0.26; 10 PIMPLE iter.; 5 PISO iter. 200
8.3 Initial residual of the pressure equation and linear solver iteration distribution along two time steps; KCS in head regular wave; Fr=0.26; $\varepsilon_{\text{err}}^{\text{SPIM}} = 0.1$; 5 PISO iter. 202
8.4 Initial residual of the pressure equation and linear solver iteration distribution along two time steps; KCS in head regular wave; Fr=0.26; $\varepsilon_{\text{err}}^{\text{SPIM}} = 1$; 5 PISO iter. 203
8.5 Initial residual of the pressure equation and linear solver iteration distribution along two time steps; KCS in head regular wave; Fr=0.26; $\varepsilon_{\text{err}}^{\text{SPIM}} = 1$; $\rho\text{err}^{\text{FSI}} = 0.001$; 5 PISO iter. 204
8.6 Time history of the pressure resistance and viscous resistance coefficients using various PIMPLE controllers; KCS Fr = 0.26; regular head wave $\lambda = 0.65L_{PP}$; steepness $\varepsilon = 1.6\%$. 205
8.7 Time history of the heave and pitch using various PIMPLE controllers; KCS Fr = 0.26; regular head wave $\lambda = 0.65L_{PP}$; steepness $\varepsilon = 1.6\%$ 205

List of Tables

3.1 Modeled quantities evaluated with VOF-RANSE resolution 43
3.2 Synthesis of notations used for the spatial discretization of a field ψ at a cell P . 46
3.3 Numerical fields updated by the FSI resolution 48
3.4 Numerical fields updated by the VOF resolution 57
3.5 Numerical fields updated by the RANSE resolution 65
3.6 Numerical fields updated by the turbulence model resolution 70
3.7 Tested temporal time schemes along the numerical studies. 85
3.8 Tested cell-centered gradient schemes along the numerical studies. 85
3.9 Tested surface normal gradient schemes along the numerical studies. 85
3.10 Tested matrix-system resolution parameters along the numerical studies. 86
3.11 Tested PIMPLE and PISO parameters along the numerical studies. 86

4.1 Test matrix for Taylor-Green time-convergence study with `U256x256` mesh. . 91
4.2 `Regular` meshes used on the Taylor-Green vortices case. 94
4.3 `Stretch` meshes used on the Taylor-Green vortices case. 95
4.4 Influence of the mesh deformations on the mesh quality 102

5.1 Regular wave characteristics. 110
5.2 Boundary conditions used for wave propagation in periodic domain. 110
5.3 Numerical parameters studied with regular wave propagation in periodic domain. 112
5.4 Characteristics of refinement zones used composing non-uniform meshes used for the regular wave propagation in periodic domain. 115
5.5 Studied configurations for the regular wave propagation in periodic domain. . . 116
5.6 Optimal aspect ratio for various wave steepness. 135

6.1 Regular wave characteristics. 143
6.2 Boundary conditions used for wave propagation with relaxation zones. 144

6.3 Numerical parameters studied with regular wave propagation with relaxation zones. 145
6.4 Studied configuration of meshes for the regular wave propagation with relaxation zones. 146
6.5 Optimal aspect ratio for various wave steepness with relaxation zone and Navier-Stokes model. 151

7.1 Main particulars of KCS geometry 168
7.2 Physical parameters for KCS experiments and numerical simulations 168
7.3 Degrees of freedom during KCS experiments and numerical simulations 168
7.4 Regular Waves conditions for KCS model scale hull 169
7.5 Boundary conditions used for naval simulation in head waves. 170
7.6 Numerical configuration parameters for the KCS in head regular wave. 171
7.7 Mesh information for KCS in calm water cases 174
7.8 Calm Water results comparison 175
7.9 Mesh information for KCS in head regular waves 175
7.10 Ratio H^{CFD}/H at the CoG 177
7.11 KCS Total resistance results, (FSS) exponent stands for *foamStar-SWENSE* and (FS) exponent for *foamStar* solver 180
7.12 Irregular sea state (SS6) characteristics 182
7.13 Improved numerical configuration parameters for the KCS in head regular wave 187
7.14 Ratio H^{CFD}/H at the CoG in the case **C1** using the improved configuration . . 189

8.1 Computational cost of simulations using various PIMPLE controller methods and with a maximum of 10 PIMPLE iterations per time step. 204

CHAPTER 1

INTRODUCTION

Seakeeping is a predominant issue for engineering offices, shipyards, and classification companies. The predictions of motions in waves and additional resistance due to waves are key elements of ship design. It comes into play for performance analyses, ship consumption optimization in order to limit greenhouse gas emissions, and equipment and personnel safety improvement. Both experimental and numerical studies can be used for ship-performance evaluation. Nowadays, naval architects are increasingly using numerical simulations for ship design and experimental data are largely used as references for numerical validation. Two main types of numerical methods can be distinguished. The first category is the low-fidelity one. This category encompasses methods based on simplified mathematical model for which analytic solutions exist or which can be numerically solved with small computational cost. The second category is the high-fidelity one. High-fidelity methods require the use of costly numerical solver but they provide an higher accuracy in modeling complex flows, with no simplification of the mathematical model. The term Computational Fluid Dynamics (CFD) is commonly used to name the high-fidelity methods.

First industrial application of CFD in the marine industry was the evaluation of calm water resistance. Thanks to advances in computer technology, and the improvement of numerical solvers, the CFD is now commonly used for problems with waves such as seakeeping and in particular the evaluation of added resistance. Today, for naval industry, the objective is to get efficient numerical models able to simulate the ships motions and loads from the calm water condition to complex irregular sea-states condition. The work done by Sigmund and Peric (2018) and Kim et al. (2022b) are some examples of what is at stake in this PhD. In order to meet the objective, one can identify three key points for the evaluation of computational methods. The first one is the efficiency of the codes for both ship motion and sea-state modeling. The second one is the adequacy of the software's capabilities with the industry's needs. And the last one is the need for clear recommendations and procedures for targeted applications.

1.1 Previous and related researches

The definition of the scope of this Ph.D. starts by the choice of the numerical solvers. This choice is motivated by the industrial context of the present work and by the previous and related research in the fields of added resistance evaluation and seakeeping. A brief introduction to the mathematical and numerical models from the literature is given in the following. Besides, an overview of the solver used along this Ph.D. is done. Finally, this section ends with specifications on the Verification and Validation process needed for the evaluation of the selected mathematical model and numerical method.

1.1.1 Choice of mathematical models

The first step towards numerical simulation is the choice of a mathematical model describing the physics of the studied problem. In naval engineering the models used are derived from Navier-Stokes equations. The first mathematical model used for seakeeping, calm water resistance and added resistance was the potential flow theory. Numerical methods based on the potential flow theory were developed from the 1960's, they fall into the low-fidelity category.

Nowadays, the potential flow theory is still used for a wide range of applications: added resistance (see e.g. Simonsen et al., 2013; Hizir et al., 2019; Zhang and El Moctar, 2019); seakeeping and maneuvering (Skejic and Faltinsen, 2008; Subramanian and Beck, 2015; Kianejad et al., 2019, see e.g.); self propulsion (Mofidi et al., 2018, see e.g.) ...

For instance, due to its low computational cost, the potential flow method is suitable for hull optimization process. In fact, optimization process leads to numerous simulations so the exclusive use of costly CFD methods is yet generally too expensive for industrial applications. Some methods use potential flow solvers based on boundary element methods with optimization algorithms (see e.g. Feng et al., 2022). Some other methods, coupling high and low-fidelity codes, benefit from the reduced computational cost of potential flow models using it as the low-fidelity component in the optimization process (see e.g. Serani et al., 2021).

For both model and full-scale ships, the flow around the hulls is turbulent. So, for high-fidelity naval simulations, the Reynolds Averaged Navier Stokes (RANS) model is commonly used. This model allows more reliable predictions thanks to the direct evaluation of viscous drag for the ship resistance problem and the inclusion of additional physical phenomena linked to viscosity such as the influence of vorticity or the interactions between the viscous wake and the waves. The first solvers based on RANS Equations (RANSE) have been developed in the 1990's, initially for the ship resistance problem in calm water (see e.g. Kodama et al., 1994),

and later in the 2000's for the seakeeping problem (see e.g. Hino, 2005).

An alternative model called Large Eddy Simulation (LES) is sometimes used in order to model accurately turbulent phenomena in the air and massively separated flows. As the LES simulation imposes very high computational cost, the Detached-Eddy Simulation (DES) method can be used. It combinesthe LES method far from the body and the RANS method near to it (see e.g. Spalart, 2009). It is employed for instance for accurate ship airwake computations (see e.g. Forrest and Owen, 2010). The application of Navier-Stokes equations without specific models for the turbulence is called the Direct Numerical Simulation (DNS). In such a case, the cell size must be sufficiently small to track all vortices. For naval applications, this method would imply an enormous and unachievable computational cost.

For the present work, because of the objectives and the physical phenomena of interest, the RANS model applied to multi-phase incompressible flow has been chosen.

1.1.2 Choice of the numerical solver

For the numerical resolution of the RANSE, several methods of discretisation are existing. The two principal are the Finite Difference (FDM) and the Finite Volume Methods (FVM). A third one is the Finite Element Method (FEM) that is nowadays not often used for RANSE resolution. Over the past few years, FVM has become the standard of CFD for naval applications. This method appears to be well adapted for complex flow simulations around offshore structures or ships mixing accuracy and reasonable computational cost. Major commercial CFD codes such as Simcenter STAR-CCM+®, Ansys fluent® or Fine/Marine® are based on FVM. For academic research but also more and more in the industry, the open-source library solver OpenFOAM® is widely used. This open-source library is developed since the beginning of the 2000's (OpenFOAM, 2022b; Jasak et al., 2007b). Different versions of this solver are available. The two main structures providing releases of OpenFOAM® are OpenCFD Limited and OpenFOAM Foundation. The main developer and manager of OpenFoam® on behalf of the OpenFOAM Foudation is CFD Direct. Bureau Veritas Marine&Offshore (BV-M&O) and the research Laboratory in Hydrodynamics, Energetics and Atmospheric Environment (LHEEA) have co-developed for a few years a solver named *foamStar* based on the OpenFOAM® library. The current version of *foamStar* is based on the OpenFOAM-Foundation OpenFOAM-5 release.

OpenFOAM software integrates different numerical solvers. *foamStar* is based on the two-phase incompressible-flow RANSE solver *interDymFoam*. This solver is widely used in academic research for naval applications with unsteady multi-phase flows. With

interDymFOAM, the multi-phase flow model is based on the Volume-Of-Fluid (VOF) method proposed by Hirt and Nichols (1981). Recent works are done using solvers based on *interDymFoam* [1]. To name but a few, Hu et al. (2016) deals with numerical wave tank and wave-structure interaction, Larsen and Fuhrman (2018) addresses turbulence models and wave propagation, Li et al. (2018a) is about wave-structure interaction, Larsen et al. (2019) addresses the accuracy of *interFoam* for wave-propagation simulations.

To fullfill the objectives the solver has to allow accurate wave generation and body motions calculations. These features are provided by *foamStar*.

Wave generation Efficient seakeeping analyses and added resistance evaluation require efficient numerical wave generation. The forcing/relaxing zone method is implemented in *foamStar*. Its implementation is based on the *wave2foam* library (Jacobsen et al., 2012). Waves profile can be generated with several potential flow models. Analytical regular waves are directly computed in *foamStar*'s code using the stream-function theory (Rienecker and Fenton, 1981) and irregular sea-states are generated using external tools *HOS-NWT* (Ducrozet et al., 2012) and *HOS-Ocean* (Ducrozet et al., 2016) developed by the LHEEA. These tools are based on the High-Order-Spectral (HOS) method (West et al., 1987; Dommermuth and Yue, 1987).

The evaluation of the accuracy of *foamStar* for wave generation has been done by Choi et al. (2020) and more recently by Kim et al. (2022a).

Body motions and wave structure interactions The coupling of the fluid solver with 6 Degrees of Freedom (DoF) body motion exists in OpenFOAM-5 but some modifications and improvements have been done in *foamStar* to take into account hydro-elastic fluid-structure interactions for slamming, springing, and whipping study (see e.g. Seng, 2012; Seng et al., 2014; Benhamou et al., 2018). That *foamStar* 6 DoFs solver is also well adapted for seakeeping studies and wave structure interactions (see e.g. Seng et al., 2017; Kim, 2021; Aliyar et al., 2022). For the present work, the bodies are assumed to be rigid.

SWENSE method and foamStar-SWENSE For naval simulation with waves, another method has to be distinguished from standard RANSE resolution. This method is based on the Spectral Wave Explicit Navier-Stokes Equations (SWENSE). It consists of decomposing RANSE into an incident part and a complementary part. The incident part represents the

1. or *interFoam* that is a twin solver. *interDymFoam* and *interFoam* are joined under the name *interFOAM* for recent OpenFOAM versions

incident nonlinear wave field based on an inviscid flow model solving the potential flow equations. The complementary part encompasses all the radiation and diffraction phenomena including the viscous flow effects. Thanks to this decomposition, the solution of the incident part of the equations is obtained by potential flow theory with already mentioned stream functions and HOS methods. Only the complementary part is solved using CFD methods. *foamStar-SWENSE* is the solver based on *foamStar* using the SWENSE method. The benefit of such a method is that the discretization of the simulation domain is only dedicated to the solution of the complementary field and even with coarse meshes, the incident field propagates efficiently. Consequently, mesh refinement far from the body can be coarser than the one required by a standard RANSE implementation. This leads to a significant reduction in computational cost. Works on SWENSE implementation into CFD solvers have been done in the past on single-phase FDM solver named ICARE-CFD (Ferrant et al., 2003; Luquet et al., 2007; Monroy et al., 2011; Reliquet et al., 2014). More recently, the SWENSE method was first and partly adapted to the two-phase FVM solver by Vukcevic (2016) in foam-extend 3.1 (2004), an add-on library of OpenFOAM. The implementation of *foamStar-SWENSE* was carried out by Li (2018). Choi (2019), (Li et al., 2021) studied the efficiency of *foamStar-SWENSE* for regular wave generation and fluid-structure interactions in regular waves without forward speed. Irregular wave propagation, naval seakeeping with 2 DoFs, and added resistance studies have been successfully performed by Kim (2021) using Diagonally Implicit Runge-Kutta (DIRK) schemes implemented by him in both *foamStar* and *foamStar-SWENSE*.

1.1.3 Evaluation of the performances of a solver

In order to evaluate both the mathematical models and numerical methods considered in the thesis, several Verification and Validation processes exist in the literature. Firstly, the selection of mathematical models depends on the studied physics, so, considering specific applications, some specificity in the models influence their validity a lot. For instance, with the RANSE model, several turbulence models are available and for a given study, it is necessary to select the most appropriate through validation procedures. As the models are coded in a numerical solver, the method of implementation impacts the accuracy of the simulation. Even with a theoretically valid physical model, it is necessary to verify the numerical implementation and the case-designing method. So, hand in hand with the validation, the verification process has to be conducted. This Verification and Validation procedure (V&V) is a keystone of CFD. For numerical simulations, V&V procedures have been discussed over the last three decades. Several approaches have been established, depending on the factors of errors taken into account

and the used statistical models. One of the pioneering article of the definition and application of V&V procedures for naval CFD was Coleman and Stern (1997). This work was motivated by the growth of interest in editorial policies of scientific journals or international research organizations about the quantification of uncertainties and errors in CFD. The objective was to reach a more rigorous methodology for the accuracy assessment, see Celik (1993). Another significant actor in the development of V&V standards is Patrick J. Roache. He was one of the first to clarify the methodologies of Verification and Validation (Roache, 1998). Nowadays, a standard validation process can be found in McHale et al. (2009) with additional commonly used simplifications. Various recent discussions on this process can be found in the literature, the V&V procedure proposed by Eça and Hoekstra (2014) is frequently used for naval CFD.

1.2 Present Contributions

This Ph.D. is part of the research conducted by the Bureau Veritas Marine&Offshore and Ecole Centrale Nantes. The objective of the present work is to develop solvers and methodologies in order to improve computational time and accuracy with regard to seakeeping and added resistance.

foamStar algorithm exploration The first contribution is the production of a synthetic workflow of the algorithm of *foamStar*. This workflow refers to the expanded implemented equations, and the structure of the implemented discretization is detailed. The corresponding section is expected to be the initial support for a clear documentation that would be used by the co-workers from of BV-M&O, BVS-M&O and the LHEEA.

Numerical improvement The present work proposes an extension of a numerical method dedicated to the resolution of the VOF equations. This development allows the use of second-order backward schemes for VOF convection. This improvement raises of the accuracy and/or numerical stability of *foamStar* for wave propagation and ship-wave interactions.

Best practices for industrial naval applications A progressive step by step analysis of *foamStar* accuracy is conducted all along the numerical studies reported in this document. This leads to the definition of best practices for naval simulations in waves matching with the industrial expectation for accuracy and computational cost.

Assement of *foamStar* accuracy for naval simulation in head wave A preliminary study of the solver performances simulating ship motions in head regular and irregular waves is

performed.

1.3 Thesis outline

Chapter 2 This chapter presents the mathematical models describing the physics that is simulated with *foamStar*. The two-phase RANSE and SWENSE are described as well as the body motions and the wave generation models.

Chapter 3 This chapter details the numerical implementation of *foamStar*. The numerical schemes necessary to the resolution are described. Then, the *foamStar* algorithm solving the set of equations exposed in Chapter 2 is explained and a representation of the *foamStar* flow-chart is built. Finally, the implementation of a second-order backward time scheme dedicated to the VOF resolution is presented. At the end of this chapter the reference numerical set-up studied in this document is defined.

Chapter 4 This chapter presents a study on Taylor-Green vortices simulation with single-phase Navier-Stokes model (so without turbulent models). An analysis is done on the accuracy of *foamStar* with the reference numerical set-up and different time schemes. The study also focuses on the influence of some mesh deformations generating stretched, non-orthogonal and skewed cells.

Chapter 5 This chapter presents a study of the accuracy of *foamStar* simulating nonlinear regular wave propagation in periodic bi-dimensional domain of one wavelength. Simulations are done using two-phases VOF-Navier-Stokes equations. The newly implemented second order backward scheme is compared to the existing Crank-Nicolson one. Various configurations of mesh refinements are compared and some recommendations are made on how to accurately simulate regular wave propagation with *foamStar*.

Chapter 6 This chapter presents a study of nonlinear regular wave propagation in a bi-dimensional long domain of several wavelengths with relaxation zones. The newly implemented second-order backward scheme is once again compared to the existing Crank-Nicolson one and the recommendations made in Chapter 5 are verified. To reach progressively the numerical configuration dedicated to naval simulation, two-phases VOF Reynolds-Averaged Navier-Stokes model and moving reference frame are considered.

Chapter 7 This chapter presents some analysis on the accuracy of *foamStar* and *foamStar-SWENSE* simulating scale model KRISO container ship moving in head regular

and irregular waves. This chapter is a preliminary study intending to asses the ability of both *foamStar* and *foamStar-SWENSE* to compute ship motion and wave added resistance in head waves.

Chapter 8 This chapter connects the work done during this Ph.D. and the industrial needs. It describes some actions done in order to ease the industrialization of *foamStar*.

Chapter 9 This chapter concludes the present work. The results obtained throughout this Ph.D. are put in perspective with various current and potential future works.

PART I

Theoretical and numerical modeling

This part presents the background of the works detailed in the rest of this document. The first section deals with the mathematical models, while the second section focuses on the numerical discretization methods.

CHAPTER 2

MATHEMATICAL MODELS

2.1 Two phase incompressible flow

2.1.1 Single-phase Navier-Stokes equations

In a fluid domain, the local Navier-Stokes (NS) equations in Lagrangian form are expressed in a Galilean reference frame $\mathscr{R}_0$ as:

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} + \rho\nabla\bullet\mathbf{u} = 0 \tag{2.1a}$$

$$\frac{\mathrm{d}(\rho\mathbf{u})}{\mathrm{d}t} = \rho\mathbf{g} + \nabla\bullet\overline{\overline{\Sigma}} \tag{2.1b}$$

with $\mathbf{u}$ the fluid velocity vector, ρ the density, $\mathbf{g}$ the gravitational acceleration vector, and $\overline{\overline{\Sigma}}$ the stress tensor. Eq. 2.1a is the continuity equation and Eq. 2.1b is the momentum equation. For a Newtonian fluid with the Stokes hypothesis, the viscous stress tensor is:

$$\overline{\overline{\Sigma}} = -\left(P + \frac{2}{3}\mu\nabla\bullet\mathbf{u}\right)\overline{\overline{\mathrm{I}}} + \mu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right) \tag{2.2}$$

with μ the dynamic viscosity coefficient and P the total pressure. Defining the dynamic pressure p_d, at a position $\mathbf{x}$ and time t,

$$p_\mathrm{d}(\mathbf{x},t) = P(\mathbf{x},t) - \rho(\mathbf{x},t)\mathbf{g}\bullet\mathbf{x} \tag{2.3}$$

the resulting NS equations under Eulerian conservative form are:

$$\frac{\partial\rho}{\partial t} + \nabla\bullet(\rho\mathbf{u}) = 0 \tag{2.4a}$$

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla\bullet(\rho\mathbf{u}\mathbf{u}) = -\nabla p_\mathrm{d} - \frac{2}{3}\nabla(\mu\nabla\bullet\mathbf{u}) - \mathbf{g}\bullet\mathbf{x}\nabla\rho + \nabla\bullet\left(\mu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)\right) \tag{2.4b}$$

2.1.2 Two-phase single-field Navier-Stokes equations

The managment of multiphase flow can be done with several numerical methods and mathematical models. The formulation adopted is the single-field volume of fluid (VOF) method (Hirt and Nichols, 1981) for the studies presented in this document. Details of such a formulation and discussions about multi-phase formulations can be found in Rusche (2002) and Marschall (2011).

The water volume fraction α is defined with $\alpha = 1$ in water phase and $\alpha = 0$ in air phase. The interface between water and air is a continuous transition zone with $0 < \alpha < 1$. The local density ρ, kinematic viscosity ν and dynamic viscosity μ are defined as:

$$\rho = \alpha\rho_{\mathrm{w}} + (1-\alpha)\rho_{\mathrm{a}} \tag{2.5a}$$
$$\mu = \alpha\mu_{\mathrm{w}} + (1-\alpha)\mu_{\mathrm{a}} \tag{2.5b}$$
$$\nu = \frac{\mu}{\rho} \tag{2.5c}$$

with 'w' and 'a' subscripts referring to water and air fields, respectively.

The single-field hypothesis is that velocity of air and water are equal in transition zones ($0 < \alpha < 1$), this unique velocity is denoted with $\mathbf{u}$. Averaging the continuity equations in both air and water phase (Rusche, 2002) and writing them in Eulerian form, it results the following equations:

$$\frac{\partial(\alpha\rho_{\mathrm{w}})}{\partial t} + \nabla\bullet(\alpha\rho_{\mathrm{w}}\mathbf{u}) = 0 \tag{2.6a}$$
$$\frac{\partial\left([1-\alpha]\rho_{\mathrm{a}}\right)}{\partial t} + \nabla\bullet([1-\alpha]\rho_{\mathrm{a}}\mathbf{u}) = 0 \tag{2.6b}$$

Considering both the air and water phases as incompressible flows, dividing Eq. 2.6a by ρ_{w} and Eq. 2.6b by ρ_{a} and then summing them, the resulting continuity equation is:

$$\nabla\bullet\mathbf{u} = 0 \tag{2.7}$$

Besides, dividing Eq. 2.6a by the ρ_{w}, it becomes:

$$\frac{\partial\alpha}{\partial t} + \nabla\bullet(\alpha\mathbf{u}) = 0 \tag{2.8}$$

Due to the continuity equation (Eq. 2.7) and the formulation of ρ and μ, the resulting VOF-NS momentum equation is:

$$\boxed{\frac{\partial(\rho\mathbf{u})}{\partial t}+\nabla\bullet(\rho\mathbf{u}\mathbf{u})=-\nabla p_{\mathrm{d}}-\mathbf{g}\bullet\mathbf{x}\nabla\rho+\nabla\bullet(\mu\nabla\mathbf{u})+\nabla\mu\bullet\nabla\mathbf{u}^{T}} \quad (2.9)$$

Two-field formulation and compression term For a two-field VOF formulation, the velocity are not continuous at the interface between air and water. Then, a relative velocity $\mathbf{u}_{\mathrm{r}}$ can be defined as:

$$\mathbf{u}_{\mathrm{r}}:=\mathbf{u}_{\mathrm{w}}-\mathbf{u}_{\mathrm{a}} \quad (2.10)$$

And a mean field velocity $\mathbf{u}$ as:

$$\mathbf{u}=\alpha\mathbf{u}_{\mathrm{w}}+(1-\alpha)\mathbf{u}_{\mathrm{a}}, \quad (2.11)$$

Once again, averaging the continuity equations in both air and water phase (Rusche, 2002) with an Eulerian form and dividing by the constant ρ_{w} and ρ_{a} respectively, gives:

$$\frac{\partial\alpha}{\partial t}+\nabla\bullet\alpha\mathbf{u}_{\mathrm{w}}=0 \quad (2.12a)$$

$$-\frac{\partial\alpha}{\partial t}+\nabla\bullet[1-\alpha]\,\mathbf{u}_{\mathrm{a}}=0 \quad (2.12b)$$

Summing Eq. 2.12a to Eq. 2.12b and using Eq. 2.11 still gives the continuity equation Eq. 2.8. Besides, From Eq. 2.10 and Eq. 2.11, the following relation can be set:

$$\mathbf{u}=\mathbf{u}_{\mathrm{w}}-(1-\alpha)\mathbf{u}_{\mathrm{r}} \quad (2.13)$$

Multiplying by alpha and rearranging it becomes:

$$\alpha\mathbf{u}_{\mathrm{w}}=\alpha\mathbf{u}+\alpha(1-\alpha)\mathbf{u}_{\mathrm{r}} \quad (2.14)$$

Finally, injecting Eq. 2.14 in Eq. 2.12a gives the following volume-fraction convection law:

$$\frac{\partial\alpha}{\partial t}+\nabla\bullet(\alpha\mathbf{u})+\nabla\bullet(\alpha(1-\alpha)\mathbf{u}_{\mathrm{r}})=0 \quad (2.15)$$

With the definition of the density (Eq. 2.5a), the resulting mass conservation is:

$$\frac{\partial\rho}{\partial t}+\nabla\bullet(\rho\mathbf{u}+[(\rho_{\mathrm{w}}-\rho_{\mathrm{a}})(1-\alpha)\alpha\mathbf{u}_{\mathrm{r}}])=0 \quad (2.16)$$

As mentioned, with *interDymFOAM* and *foamStar*, the VOF formulation is a single-field one. Then, no distinction is made between water and air velocity. However, in order to limit

the smearing of the transition zone of the interface, Rusche (2003) proposed the addition of a compression term inspired by this two-field consideration. Defining:

$$\mathbf{u}_{\text{comp}} = (1-\alpha)\alpha\mathbf{u}_{\text{r}} \tag{2.17}$$

with $\mathbf{u}_{\text{r}}$ a vector field term [1], the convection equation of compressed single-field VOF becomes:

$$\frac{\partial\alpha}{\partial t} + \nabla\bullet(\alpha\mathbf{u}) + \underbrace{\nabla\bullet\mathbf{u}_{\text{comp}}}_{\textit{compression term}} = 0 \tag{2.18}$$

and the mass conservation:

$$\frac{\partial\rho}{\partial t} + \nabla\bullet\left(\rho\mathbf{u} + \left[(\rho_{\text{w}}-\rho_{\text{a}})\mathbf{u}_{\text{comp}}\right]\right) = 0 \tag{2.19}$$

Using a compression term but keeping a pure single-field formulation for the momentum equation (Eq. 2.9) results in breaking the momentum conservation in the transition zone at the interface. In OpenFOAM, this additional term is partially taken into account in the momentum equation using it as an additional convective term in an analogous way to Eq. 2.19. The resulting equation is:

$$\frac{\partial\rho\mathbf{u}}{\partial t} + \nabla\bullet\left(\rho\mathbf{u}\mathbf{u} + (\rho_{\text{w}}-\rho_{\text{a}})\mathbf{u}_{\text{comp}}\mathbf{u}\right) = -\nabla p_{\text{d}} - \mathbf{g}\bullet\mathbf{x}\nabla\rho + \nabla\bullet(\mu\nabla\mathbf{u}) + \nabla\mu\bullet\nabla\mathbf{u}^{T} \tag{2.20}$$

2.1.3 Two phase single-field Reynolds Average Navier-Stokes equations

For naval applications, the Navier-Stokes equations are often used in a transformed form called Reynolds Averaged Navier-Stokes Equations (RANSE). This model consists of using the Reynolds statistical decomposition and modelling the average turbulence effect with the Eddy-Viscosity method. In the resulting equation, the turbulent kinematic viscosity ν_{t} and the turbulent kinetic energy k are additional terms modelling the small-scale phenomena related to velocity fluctuations. From the turbulent kinematic viscosity ν_{t} and the density ρ, the effective dynamic viscosity μ_{eff} can be defined as follows:

$$\mu_{\text{eff}} = \mu + \rho\nu_{\text{t}}, \tag{2.21}$$

1. In practice $\mathbf{u}_{\text{r}}$ is used as a numerical corrector and its definition depends on the needs. (see Sec. a))

and a pressure p as,

$$p = p_{\mathrm{d}} + \frac{2}{3}\rho k \tag{2.22}$$

Then, the VOF-RANS momentum equation becomes:

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla\bullet(\rho\mathbf{u}\mathbf{u} + (\rho_{\mathrm{w}} - \rho_{\mathrm{a}})\mathbf{u}_{\mathrm{comp}}\mathbf{u}) = -\nabla p - \mathbf{g}\bullet\mathbf{x}\nabla\rho + \nabla\bullet(\mu_{\mathrm{eff}}\nabla\mathbf{u}) + \nabla\mu_{\mathrm{eff}}\bullet\nabla\mathbf{u}^T \tag{2.23}$$

This equation is convenient for naval applications because it models unsteady flow taking into account inertial, volumic, pressure and viscous forces. The small scale turbulent phenomena are only considered through their average influence, so RANSE resolution has a significantly smaller computational cost than DNS.

In the following, the symbol p is used for both Navier-Stokes and RANS models. With Navier-Stokes model, as no turbulent model are used, k is null so $p = p_{\mathrm{d}}$.

Turbulence model The closure of the RANSE system implies the choice of a turbulence model. The k-ω-SST model (Menter, 1994) is widely used in naval CFD literature. In order to stabilize the model in nearly potential flow regions, especially close to the free surface and limit the turbulent exponential growth a modified $k-\omega$ closure was proposed by Larsen and Fuhrman (2018). In *foamStar* this model is adapted to a SST formulation with the following equations:

$$\frac{\partial\rho k}{\partial t} + \nabla\bullet(\rho\mathbf{u}k) - \nabla\bullet((\mu + \rho\alpha_k\nu_{\mathrm{t}})\boldsymbol{\nabla}k) = \rho\nu_{\mathrm{t}}p_0 - \frac{2}{3}\rho k\nabla\bullet\mathbf{u} - \rho\beta^* k\omega - \rho P_{\mathrm{b}} \tag{2.24a}$$

$$\frac{\partial\rho\omega}{\partial t} + \nabla\bullet(\rho\mathbf{u}\omega) - \nabla\bullet((\mu + \rho\alpha_\omega\nu_{\mathrm{t}})\boldsymbol{\nabla}\omega) = \gamma\rho p_0 - \frac{2}{3}\rho\gamma\omega\nabla\bullet\mathbf{u} - \rho\beta^*\omega^2 - \rho(F_1 - 1)CD_{k\omega} \tag{2.24b}$$

with,

$$\nu_{\mathrm{t}} = \frac{a_1 k}{\max(a_1\omega, b_1 F_2\, p_0)} \tag{2.25}$$

$$P_{\mathrm{b}} = \alpha_{\mathrm{b}}^*\nu_{\mathrm{t}}\boldsymbol{g}\bullet\nabla\rho \tag{2.26}$$

$$p_0 = \frac{1}{2}, \left(\nabla(\mathbf{u}) + \nabla(\mathbf{u})^T\right) : \left(\nabla(\mathbf{u}) + \nabla(\mathbf{u})^T\right) \tag{2.27}$$

$$\tag{2.28}$$

The formulations of the coefficients F_1, F_2, a_1, a_2, b_1, b_2 α_k, γ, α_ω, α_{b}^*, β^* and $CD_{k\omega}$ are detailed in Kim (2021). In the present document, this turbulent model is called fs-k-ω-SST.

2.1.4 VOF-RANSE equations with a moving referential

In *foamStar*, the earth reference frame is assumed to be Galilean. With the *foamStar* version used in the present document, the equations are written in the moving-domain reference frame $\mathscr{R}_{\mathrm{D}}$. The considered motion of the domain can only be rectilinear regarding to the earth reference frame, its velocity is noted $\mathbf{v}_0$ and its acceleration is noted $\mathbf{a}_{\mathrm{cc}_0}$. Because of the rectilinear motion, all the VOF-RANSE equations previously defined keep the same formulation in $\mathscr{R}_{\mathrm{D}}$ except the momentum equation in which an additional acceleration appears. The momentum equation in $\mathscr{R}_{\mathrm{D}}$ is:

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla\bullet(\rho\mathbf{u}\mathbf{u} + (\rho_{\mathrm{w}} - \rho_{\mathrm{a}})\mathbf{u}_{\mathrm{comp}}\mathbf{u}) = \\ -\nabla p - \mathbf{g}\bullet\mathbf{x}\nabla\rho + \nabla\bullet(\mu_{\mathrm{eff}}\nabla\mathbf{u}) + \nabla\mu_{\mathrm{eff}}\bullet\nabla\mathbf{u}^T - \rho\,\mathbf{a}_{\mathrm{cc}_0} \quad (2.29)$$

2.2 Fluid-structure interactions

The previous section describes the equations for an infinite fluid domain. In order to compute the solutions with a moving body as it is the case in naval simulations, two further aspects should be considered. Firstly various kinds of boundaries need to be defined around the fluid domain, and secondly the loads acting on the boundaries need to be evaluated together with the solid dynamics.

2.2.1 Wall boundary

For naval simulations, boundaries such as solid bodies and walls have to be modeled. The mathematical model presented before for the inner fluid is still available but an additional boundary equation is added. Given a fluid domain Ω, $\partial\Omega$ defines the physical non-porous boundary of the domain and $\mathbf{u}_{\mathrm{wall}}$ defines the local wall velocity. Keeping $\mathbf{u}$ the fluid velocity, with the hypothesis of adherence of the fluid to the wall results in:

$$\forall M \in \partial\Omega,\ \mathbf{u}(M) = \mathbf{u}_{\mathrm{wall}}(M) \quad (2.30)$$

2.2.2 Rigid body motions

On the body side, for walls moving with a prescribed velocity, nothing is to solve and $\mathbf{u}_{\text{wall}}$ is imposed. However, for naval simulations, the ship motions have to be modeled. In the present work the free bodies are assumed to be rigid and moving with up to six degrees of freedom (6 DoFs). To represent the ship's motions, an orthogonal Galilean reference frame $\mathscr{R}_0$ = (O, $\mathbf{e_x}$, $\mathbf{e_y}$, $\mathbf{e_z}$) [2] and an orthonormal boat reference frame $\mathscr{R}_b$ (C_g, $\boldsymbol{\varepsilon_x}$ $\boldsymbol{\varepsilon_y}$, $\boldsymbol{\varepsilon_z}$) are defined. The directions of $\boldsymbol{\varepsilon_x}$, $\boldsymbol{\varepsilon_y}$, $\boldsymbol{\varepsilon_z}$ are fixed regarding to the boat orientation. C_g is the center of gravity of the ship. The rotations angles of the ship are the roll, pitch and yaw Tait-Bryan Euler angles, respectively noted ϕ, θ, ψ. Figure 2.1 illustrates these coordinate systems.

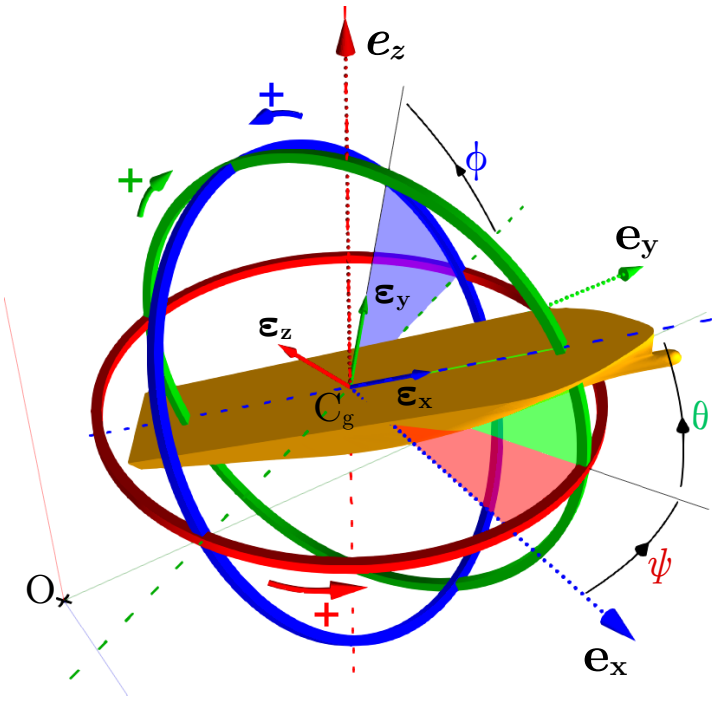


Figure 2.1 – Representation of the roll, pitch and yaw (ϕ, θ, ψ) rotations of boat referential (C_g, $\boldsymbol{\varepsilon_x}$, $\boldsymbol{\varepsilon_y}$, $\boldsymbol{\varepsilon_z}$) regarding to Galilean referential (O, $\mathbf{e_x}$, $\mathbf{e_y}$, $\mathbf{e_z}$)

The transformation matrix from the base $\mathscr{B}_0$ = ($\mathbf{e_x}$, $\mathbf{e_y}$, $\mathbf{e_z}$) to the base $\mathscr{B}_b$ = ($\boldsymbol{\varepsilon_x}$, $\boldsymbol{\varepsilon_y}$, $\boldsymbol{\varepsilon_z}$)

2. In the current *foamStar* implementation, for a moving referential, the acceleration is not taken into account during the body motion resolution. The reference frame $\mathscr{R}_D$ is assimilated to a Galilean reference frame $\mathscr{R}_0$. Consequently the body motion calculation is only valid out off acceleration phases.

is:

$$\mathbf{R}_0^{\mathrm{b}} = \begin{bmatrix} \cos\psi\cos\theta & -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\cos\phi\sin\theta \\ \sin\psi\cos\theta & \cos\psi\cos\phi + \sin\phi\sin\theta\sin\psi & -\cos\psi\sin\phi + \sin\phi\cos\phi\sin\theta \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{2.31}$$

and, from $\mathscr{B}_b$ to $\mathscr{B}_0$:

$$\mathbf{R}_{\mathrm{b}}^0 = \mathbf{R}_0^{\mathrm{b}^{-1}} = \mathbf{R}_0^{\mathrm{b}^T} \tag{2.32}$$

Consequently, given a vector $\mathbf{v}$ expressed in $\mathscr{B}_0$ coordinates and the same vector $\mathbf{v}^{\mathrm{b}}$ expressed in $\mathscr{B}_b$ coordinates,

$$\mathbf{v}^{\mathrm{b}} = \mathbf{R}_0^{\mathrm{b}}\mathbf{v} \tag{2.33}$$

Given a point M, its position in $\mathscr{R}_0$ is $\mathbf{X}_{\mathrm{M}} = \mathbf{OM}$ and $\boldsymbol{\xi}_{\mathrm{M}}^{\mathrm{b}} = \mathbf{C_g M}^{\mathrm{b}}$ in $\mathscr{R}_b$. $\mathbf{X}_{\mathrm{M}}$ and $\boldsymbol{\xi}_{\mathrm{M}}^{\mathrm{b}}$ are linked by the following relation:

$$\boldsymbol{\xi}_{\mathrm{M}}^{\mathrm{b}} = \mathbf{R}_0^{\mathrm{b}}\left(\mathbf{X}_{\mathrm{M}} - \mathbf{X}_{\mathrm{Cg}}\right) \tag{2.34}$$

with $\mathbf{X}_{\mathrm{Cg}}$ the position of $\mathrm{C_g}$ which is $\mathbf{OC_g}$ expressed in $\mathscr{R}_0$. The velocity of $\mathrm{C_g}$ in $\mathscr{R}_0$ is:

$$\dot{\mathbf{X}}_{\mathrm{Cg}} = \left.\frac{\mathrm{d}\mathbf{X}_{\mathrm{Cg}}}{\mathrm{d}t}\right|_{\mathscr{R}_0} \tag{2.35}$$

and its acceleration is:

$$\ddot{\mathbf{X}}_{\mathrm{Cg}} = \left.\frac{\mathrm{d}\mathbf{X}_{\mathrm{Cg}}^2}{\mathrm{d}t^2}\right|_{\mathscr{R}_0} \tag{2.36}$$

The rotation vector of $\mathscr{R}_b$ regarding to $\mathscr{R}_0$ and expressed in $\mathscr{B}_b$ is noted $\boldsymbol{\omega}^{\mathrm{b}}$. Its derivation in the $\mathscr{R}_{\mathrm{b}}$ is:

$$\dot{\boldsymbol{\omega}}^{\mathrm{b}} = \left.\frac{\mathrm{d}\boldsymbol{\omega}^{\mathrm{b}}}{\mathrm{d}t}\right|_{\mathscr{R}_{\mathrm{b}}} \tag{2.37}$$

The Euler angles are regrouped as a vector:

$$\boldsymbol{\Theta} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \tag{2.38}$$

and the variation rate of these angles is noted:

$$\dot{\mathbf{\Theta}} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2.39}$$

The relation between $\boldsymbol{\omega}^{\mathrm{b}}$ and $\dot{\mathbf{\Theta}}$ is as follows:

$$\dot{\mathbf{\Theta}} = \mathbf{T}\boldsymbol{\omega}^{\mathrm{b}} \quad \text{with} \quad \mathbf{T} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \tag{2.40}$$

Using the introduced notations, the Newton second law can be written in the Galilean reference frame $\mathscr{R}_0$. With the body mass m, the relation between the linear momentum and the external forces is:

$$m\ddot{\mathbf{X}}_{\mathrm{Cg}} = m\mathbf{g} + \mathbf{F}_{\mathrm{fluid}} + \mathbf{F}_{\mathrm{add}} + \mathbf{F}_{\mathrm{c}} \tag{2.41}$$

where $\mathbf{F}_{\mathrm{fluid}}$ are the fluid forces, $\mathbf{F}_{\mathrm{add}}$ are additional forces depending on the loads taken into account, $\mathbf{F}_{\mathrm{c}}$ are the constraint forces relative to the imposed motions. The expression of these constraints in *foamStar* is detailed in Section 3.3.1.

Defining $\mathbf{I}_{\mathrm{g}}$ the inertia matrix of the boat in $\mathscr{R}_{\mathrm{b}}$, the relation between the angular momentum and the external moments is:

$$\mathbf{I}_{\mathrm{g}}\dot{\boldsymbol{\omega}}^{\mathrm{b}} = \mathbf{M}^{\mathrm{b}}_{\mathrm{fluid}} + \mathbf{M}^{\mathrm{b}}_{\mathrm{add}} + \mathbf{M}^{\mathrm{b}}_{\mathrm{c}} + \boldsymbol{\omega}^{\mathrm{b}} \wedge \left(\mathbf{I}_{\mathrm{g}}\boldsymbol{\omega}^{\mathrm{b}}\right) \tag{2.42}$$

where $\mathbf{M}^{\mathrm{b}}_{\mathrm{fluid}}$, $\mathbf{M}^{\mathrm{b}}_{\mathrm{add}}$, $\mathbf{M}^{\mathrm{b}}_{\mathrm{c}}$ are respectively the fluid, additional and constraint moments about $\mathrm{C_g}$ expressed in $\mathscr{B}_b$.

Defining the body wall $\delta\Omega_{body}$, the formulations of the fluid force and moments in $\mathscr{R}_{\mathrm{b}}$ are:

$$\mathbf{F}_{\mathrm{fluid}} = \int_{\delta\Omega_{body}} \mathbf{dS} \bullet \overline{\overline{\Sigma}} \tag{2.43a}$$

$$\mathbf{M}^{\mathrm{b}}_{\mathrm{fluid}} = \mathbf{R}^{\mathrm{b}}_{0}\left[\int_{M\in\delta\Omega_{body}} \mathbf{dS} \bullet \overline{\overline{\Sigma}} \wedge \mathbf{C_g M}\right] \tag{2.43b}$$

with,

$$\overline{\overline{\Sigma}} = -\left[p + \rho \mathbf{g} \bullet \mathbf{x}\right] \overline{\overline{\mathrm{I}}} + \mu_{\mathrm{eff}} \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right) \tag{2.44}$$

Finally, because of the rigid motions, the body wall velocity $\mathbf{u}_{\mathrm{wall}}$ is related to $\mathbf{X}_{\mathrm{Cg}}$ and $\boldsymbol{\omega}^{\mathrm{b}}$ by:

$$\forall M \in \partial\Omega_{body}, \quad \mathbf{X}_{\mathrm{M}} = \mathbf{X}_{\mathrm{Cg}} + \mathbf{R}_{\mathrm{b}}^{0} \left(\mathbf{C}_{\mathbf{g}} \mathbf{M}^{\mathrm{b}}\right) \tag{2.45a}$$

$$\mathbf{u}_{\mathrm{wall}}(M) = \dot{\mathbf{X}}_{\mathrm{Cg}} + \mathbf{R}_{\mathrm{b}}^{0} \left[\mathbf{C}_{\mathbf{g}} \mathbf{M}^{\mathrm{b}} \wedge \boldsymbol{\omega}^{\mathrm{b}}\right] \tag{2.45b}$$

With this last equation, the FSI system is closed.

2.3 Wave generation

2.3.1 Wave model

a) Regular wave model

For naval simulations with waves, without wavemaker the generation of wave fields often needs analytical solutions. For regular waves modeling, the widely used model is the stream-function theory established by Rienecker and Fenton (1981). In the following, the stream-function model is briefly described using the notations from Ducrozet et al. (2019).

Given a 2D periodic regular wave with a constant phase velocity c a period T and a wavelength λ, defining a reference frame $\mathscr{R}_w$ moving at c, in the direction of the wave propagation, the wave profile is constant in $\mathscr{R}_w$. Assuming an irrotational divergence-free flow with a velocity field $\mathbf{u}(x,z) = (u(x,z),\ w(x,z))$ in $\mathscr{R}_w$, a stream-function $\psi(x,z)$ can be defined as:

$$u = \frac{\partial \psi}{\partial z}, \quad w = -\frac{\partial \psi}{\partial x} \tag{2.46}$$

Within potential flow theory, the irrotational hypothesis implies the Laplace equation:

$$\Delta \psi = 0 \tag{2.47}$$

and the Bernoulli equation is verified:

$$\frac{P}{\rho} = -gz - \frac{1}{2}\left[\left(\frac{\partial \psi}{\partial x}\right)^2 + \left(\frac{\partial \psi}{\partial z}\right)^2\right] + R \tag{2.48}$$

where R is the so-called Bernoulli constant.

Defining the free surface elevation $\eta(x)$, and fixing a null pressure at the free surface, the Bernoulli equation expressed at the free surface gives the following dynamic free surface boundary condition:

$$\forall (x,z) = (x,\eta(x)), \quad g\eta + \frac{1}{2}\left[\left(\frac{\partial \psi}{\partial x}\right)^2 + \left(\frac{\partial \psi}{\partial z}\right)^2\right] = R \tag{2.49}$$

Using Fourier series decompositions, this model provides accurate low-cost resolution of the velocity and pressure fields in a single-field domain.

b) Irregular wave model

The analytical model used for irregular wave generation is the High Order Spectral (HOS) model initially developed by West et al. (1987) and Dommermuth and Yue (1987). As for the regular wave model previously described, the HOS model is a spectral decomposition non-breaking nonlinear wave fields respecting potential flow theory.

For this method, the 3D fields are defined in the fixed Galilean reference frame $\mathscr{R}_0$. The irrotational-flow hypothesis allows the definition of the potential $\phi(x,y,z,t)$ with the following equation,

$$\mathbf{u}(x,y,z,t) = \nabla \phi \tag{2.50}$$

The potential flow model implies the following equations:

$$\Delta \phi = 0 \tag{2.51}$$

$$\frac{P}{\rho} = -\frac{\partial \phi}{\partial t} - gz - \frac{1}{2}\nabla \phi^2 + \frac{P_{\text{atm}}}{\rho} \tag{2.52}$$

In Eq. 2.52, the Bernoulli constant is chosen equal to the atmospheric pressure at the free surface P_{atm}.

Defining the free surface elevation $\eta(x,y,t)$ in referential $\mathscr{R}_0$, the kinematic and normal

dynamic free surface boundary conditions are,

$$\frac{\partial \eta}{\partial t} = \frac{\partial \phi}{\partial z} - \frac{\partial \phi}{\partial x}\frac{\partial \eta}{\partial x} - \frac{\partial \phi}{\partial y}\frac{\partial \eta}{\partial y} \tag{2.53}$$

$$\frac{\partial \phi}{\partial t} = -g\eta - \frac{1}{2}\nabla\phi^2 \tag{2.54}$$

Following the method proposed by West et al. (1987), the HOS method consists in defining the following 2D field:

$$\widetilde{\phi}(x,y,t) = \phi(x,y,\eta(x,y,t),t) \tag{2.55}$$

and noting W the vertical velocity at the free surface:

$$W(x,y,t) = \frac{\partial \phi}{\partial z}(x,y,\eta(x,y,t),t) \tag{2.56}$$

With these notations, the free surface boundary conditions can be rewritten as follows:

$$\frac{\partial \eta}{\partial t} = \left(1 + \frac{\partial \eta}{\partial x}^2 + \frac{\partial \eta}{\partial y}^2\right) W - \frac{\partial \widetilde{\phi}}{\partial x}\frac{\partial \eta}{\partial x} - \frac{\partial \eta}{\partial y}\frac{\partial \widetilde{\phi}}{\partial y} \tag{2.57}$$

$$\frac{\partial \widetilde{\phi}}{\partial t} = -g\eta - \frac{1}{2}\nabla\widetilde{\phi}^2 + \frac{1}{2}\left(1 + \frac{\partial \eta}{\partial x}^2 + \frac{\partial \eta}{\partial y}^2\right) W^2 \tag{2.58}$$

Using Fourier series decompositions and Taylor expansions, this model provides accurate moderate-cost resolution of the velocity and pressure fields in a single-field domain. The open-source solvers HOS-NWT (Ducrozet et al., 2012) and HOS-Ocean (Ducrozet et al., 2016) are both based on HOS model for wave tank and open ocean simulations.

To take into account wave-breaking phenomenon, some additional models can be used. A wave-breaking criterion was defined by Barthelemy et al. (2018) and energy loss through additional viscous term was proposed by Tian et al. (2012). An implementation of these models into HOS-NWT was proposed by Seiffert et al. (2017).

c) Relaxation zone

The relaxation zone approach is used to efficiently generate and absorb waves in the computational domain (Jacobsen et al., 2012; Li et al., 2021). A generic field ψ in the relaxation

zone is defined as,

$$\psi = (1-w)\psi^{CFD} + w\psi^{\text{Target}} \tag{2.59}$$

where $0 \leq w \leq 1$ is a weight coefficient, ψ^{CFD} the flow field from the chosen CFD model, and ψ^{target} the target field. Through this formulation, the resulting field ψ is a blended solution between ψ^{CFD} and ψ^{target}. In *foamStar*, the targeted field ψ^{target} can be computed using the previously mentioned wave-models.

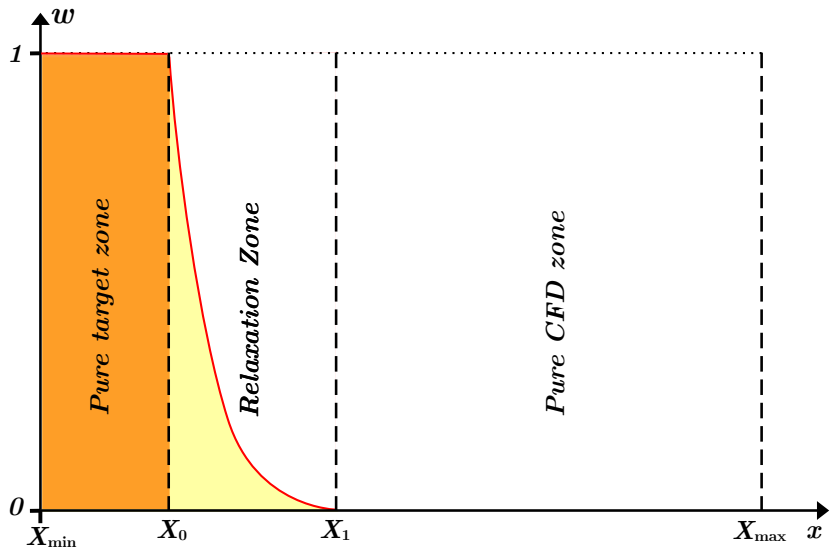


Figure 2.2 – Relaxation-zone weight (w) distribution along a simulation domain

In Figure 2.2 the red line represents the distribution of w over a computational domain. Given a 2D domain Ω in the (X,Y) plane defined by:

$$\Omega = \left\{(x,y) \in \mathbb{R}^2 | X_{\min} \leq x \leq X_{\max}, \text{ and } Y_{\min} \leq y \leq Y_{\max}\right\}$$

Then, considering a relaxation zone from X_0 to X_1, a normalized coordinate ξ is defined as:

$$\begin{array}{rcll} \xi : \Omega & \longrightarrow & [0,1] & \\ x & \longmapsto & 1 & \text{if } x \leq X_0 \\ x & \longmapsto & \dfrac{x - X_0}{X_1 - X_0} & \text{if } X_0 \leq x \leq X_1 \\ x & \longmapsto & 0 & \text{if } X_1 \leq x \end{array} \tag{2.60}$$

By default, in *foamStar*, w is an exponential weight function defined as,

$$\begin{array}{rcl} w : [0,1] & \longrightarrow & [0,1] \\ \xi & \longmapsto & \dfrac{e^{\xi^{3.5}}-1}{e-1} \end{array} \tag{2.61}$$

2.4 The Spectral Wave Explicit Navier-Stokes Equations (SWENSE)

The Spectral Wave Explicit Navier-Stokes (SWENS) method (Ferrant et al., 2003) was developed in the LHEEA in order to accurately simulate wave-structure interactions with complex sea states. The main idea is to solve the incident fields, mainly corresponding to the wave field, and the complementary fields separately (Fig. 2.3). Complementary fields are defined as the difference between the total fields and the incident ones, so they takes into account the diffracted fields but also all the corrections implied by the chosen incident fields. For example, they encompasses phenomena neglected in the incident fields and if necessary some non-physical numerical considerations related to the incident fields solving method.

Mathematically, the SWENS method is based on the hypothesis that the incident velocity and pressure fields verify the Euler equations and the total fields the RANS equations. Actually, for nonbreaking wave propagation in open field (no body presence), viscosity is negligible in the Navier-Stokes equations. These Euler equations write:

$$\nabla \bullet \mathbf{u} = 0 \tag{2.62a}$$

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \bullet (\rho \mathbf{u}\mathbf{u}) = -\nabla P + \rho \mathbf{g} \tag{2.62b}$$

The governing equations of the complementary field are called Spectral Wave Explicit Navier-Stokes equations (SWENSE).

With SWENSE, only the complementary part is solved using a CFD solver. The incident fields are already known before the simulation using a spectral approach (Sec. 2.3.1) and appeared as forcing terms during the computation. The most accurate method to get these initial fields is to compute them using potential flow theory. Those methods are discussed in the following.

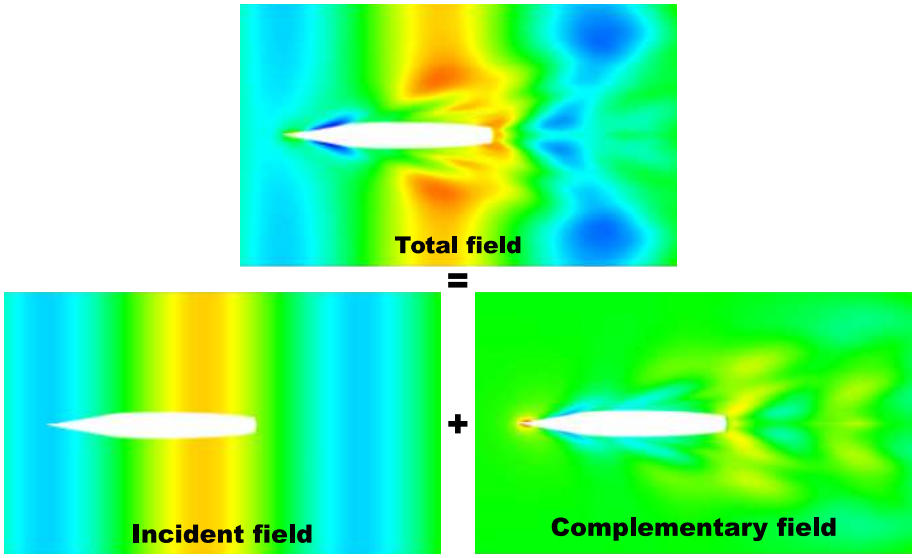


Figure 2.3 – The SWENSE method decomposes the total field into an incident part and a complementary part

Assuming the incident part of the flow is inviscid and satisfying the incompressible flow condition, the incident velocity ($\mathbf{u}_\mathrm{I}$) and the incident pressure (p_I) follow the continuity equation (Eq. 2.63a) and the Euler equation (Eq. 2.63b). The Euler equation is valid only for single-phase flow where $\rho = \rho_\mathrm{water}$.

$$\nabla \bullet \mathbf{u}_\mathrm{I} = 0 \tag{2.63a}$$

$$\frac{\partial(\rho_\mathrm{water}\mathbf{u}_\mathrm{I})}{\partial t} + \nabla \bullet (\rho_\mathrm{water}\mathbf{u}_\mathrm{I}\mathbf{u}_\mathrm{I}) = -\nabla P_\mathrm{I} + \rho_\mathrm{water}\,\mathbf{g} \tag{2.63b}$$

The SWENSE formulation subtracts the incident Euler equation (Eq. 2.63b) from the original RANS momentum equation (Eq. 2.23). As shown by Li (2018) the standard SWENSE formulation is convenient for single-phase flow but quite problematic for two-phase flow approach. Thanks to the spectral methods, the incident wave fields can be extended in the entire computational domain directly. From this extension, Euler equation (Eq. 2.63b) is valid in the entire domain. Consequently, subtracting Eq. 2.63b from Eq. 2.23 generates an extra pressure term that cannot be neglected in air and causes stability issues. To solve this issue, Li (2018) proposed the modified incident pressure using mixture density:

$$P_\mathrm{I}^* = \rho \frac{P_\mathrm{I}}{\rho_\mathrm{I}}, \tag{2.64}$$

where p_{I}^{*} is the modified incident pressure, ρ is the density of the considered fluid phase and ρ_{I} is the density of the incident field solution ($\rho_{\mathrm{I}} = \rho_{\mathrm{water}}$). The modified Euler's equation for incident terms reads:

$$\frac{\partial \mathbf{u}_{\mathrm{I}}}{\partial t} + \nabla\bullet(\mathbf{u}_{\mathrm{I}}\mathbf{u}_{\mathrm{I}}) = -\frac{\nabla P_{\mathrm{I}}^{*}}{\rho} + \frac{P_{\mathrm{I}}}{\rho_{\mathrm{I}}}\frac{\nabla\rho}{\rho} + \mathbf{g} \tag{2.65}$$

Eq. (2.66) presents the wave field decomposition. The VOF field is not decomposed in this formulation and uses the same VOF convection equation as the classic two-phase flow solver.

$$\begin{aligned} \mathbf{u} &= \mathbf{u}_{\mathrm{I}} + \mathbf{u}_{\mathrm{C}} \\ P &= P_{\mathrm{I}}^{*} + P_{\mathrm{C}} \end{aligned} \tag{2.66}$$

Using this decomposition and the continuity equation:

$$\frac{\partial \rho}{\partial t} + \nabla\bullet(\rho\mathbf{u}) = 0 \tag{2.67}$$

the Eq. 2.65 can be transform into:

$$\frac{\partial \rho\mathbf{u}_{\mathrm{I}}}{\partial t} + \nabla\bullet(\rho\mathbf{u}_{\mathrm{I}}\mathbf{u}_{\mathrm{I}}) + \mathbf{u}_{\mathrm{I}}\nabla\bullet(\rho\mathbf{u}_{\mathrm{C}}) = -\nabla P_{\mathrm{I}}^{*} + \frac{P_{\mathrm{I}}}{\rho_{\mathrm{I}}}\nabla\rho + \rho\mathbf{g} \tag{2.68}$$

Then, the subtraction of the incident flow equations from the original RANSE yields a new set of equations for the complementary flow, namely the continuity equation (Eq. 2.69) and the momentum equation (Eq. 2.70):

$$\nabla\bullet\mathbf{u}_{\mathrm{C}} = 0 \tag{2.69}$$

$$\frac{\partial(\rho\mathbf{u}_{\mathrm{C}})}{\partial t} + \nabla\bullet(\rho\mathbf{u}_{\mathrm{C}}\mathbf{u}) + \rho\mathbf{u}_{\mathrm{C}}\bullet\nabla\mathbf{u}_{\mathrm{I}} = -\nabla P_{\mathrm{C}} - \frac{P_{\mathrm{I}}}{\rho_{\mathrm{I}}}\nabla\rho + \nabla\bullet(\mu_{\mathrm{eff}}\nabla\mathbf{u}_{\mathrm{C}}) + \nabla\mathbf{u}_{\mathrm{C}}\bullet\nabla\mu_{\mathrm{eff}} - \frac{2}{3}\nabla(\rho\, k) \tag{2.70}$$

Note that the viscous term $\nabla\bullet(\mu_{\mathrm{eff}}(\nabla\mathbf{u}_{\mathrm{I}} + \nabla\mathbf{u}_{\mathrm{I}}^{\mathrm{T}}))$ is considered negligible and it is not included in Eq. (2.70). See Li (2018) for more details.

Since this incident wave field is not computed by the SWENSE solver some advantages are expected. Firstly, only the complementary field has to be damped in the outer region. This is a smaller magnitude field, easier to absorb than the total field including incident waves.

Secondly, the mesh has to be designed for the complementary field and this gives the opportunity for reducing the cells number by using a coarser resolution in the far field and consequently reducing the computation time (typically by using similar meshes as for calm water ship resistance problems.

CHAPTER 3

NUMERICAL METHODS

This chapter details the numerical implementation of the set of equations introduced in the previous chapter. The numerical models described here are those implemented in the open-source libraries OpenFOAM-5 or in one derivation implemented in the code *foamStar*. In the literature, details on the numerical implementation of OpenFOAM are provided by Jasak (1996), Darwish and Moukalled (2021) and Greenshields and Weller (2022). This chapter does not give a complete list of numerical models availlable in OpenFOAM-5 or *foamStar* but a comprehensive set of the relevant options used in this document. All those models are based on the Finite-Volume method and solved with semi-implicit schemes. The spatial discretization schemes are detailed first, then the temporal discretization scheme, and finally the full resolution algorithm.

3.1 Finite Volume spatial discretisation

Solving partial differential equations over a finite computational domain requires spatial and temporal discretization. This section gives an overview of the spatial discretization implemented in OpenFOAM.

3.1.1 Mesh characteristics

The spatial discretization method employed in OpenFOAM is the Finite Volume method. It consists in discretizing the computational domain into cells considered as elementary volumes. In OpenFOAM, the mesh has to respect some properties:

— the cells have to cover the entire numerical domain;
— cells must not overlap;
— cells have to be convex;
— cells have to be closed by an arbitrary number of faces;
— The volume of cells cannot be zero;

— A face cannot belong to more than two cells. A boundary face belong to one cell when an internal face belongs to two cells.

Each cell is identified by its geometric center. Figure 3.1 represents two contiguous cells sharing a face f where P and N are the two centers. By extension, the P and N indexes identify the cells with P and N as geometric centers respectively. V_P is the volume of the cell P, S_f is the surface of the face f and $\mathbf{n}_\mathrm{f}$ its outer normal vector. For more clarity in the following the vector $\boldsymbol{S}_\mathrm{f} = S_\mathrm{f}\mathbf{n}_\mathrm{f}$ will also be used. In the following, P index identifies the cell of interest and N index for the neighbor cells.

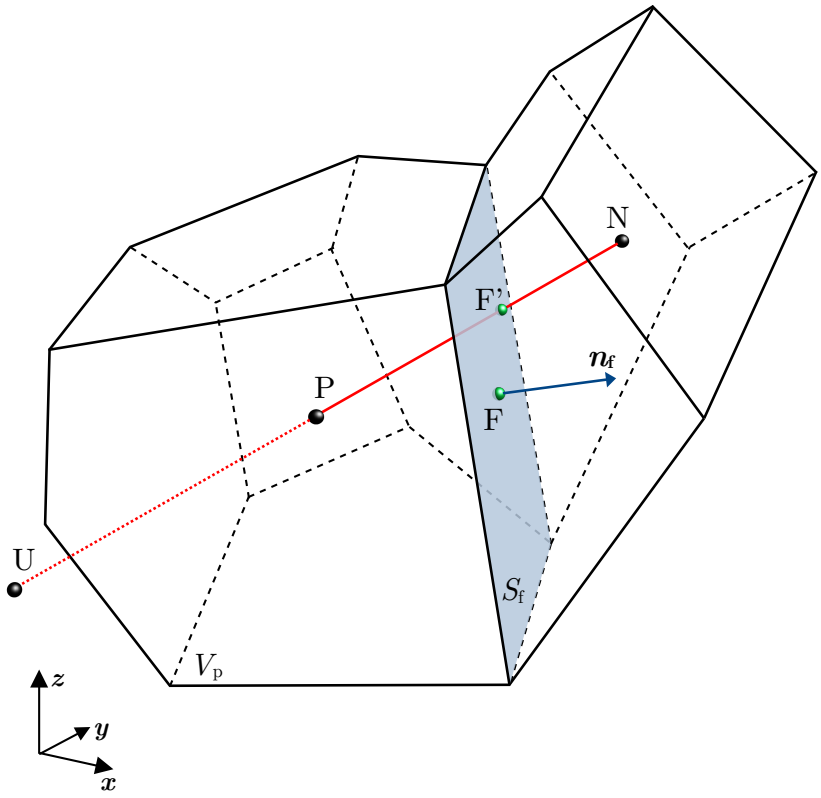


Figure 3.1 – Two neighboring control volumes sharing a face f from an unstructured 3D mesh

3.1.2 Cell and face center values

Considering a cell volume V_P with its geometric center P defined as $\int_{V_\mathrm{P}} \mathbf{PM}\, dV(M) = \mathbf{0}$, we have for a generic field ψ the following second-order approximation:

$$\iiint_{V_\mathrm{P}} \psi\, dV \approx V_\mathrm{P}\, \psi(\mathrm{P}) \tag{3.1}$$

where $\psi(\mathrm{P})$ is the exact value of ψ at P. It is important to keep in mind that this exact value $\psi(\mathrm{P})$ is not the calculated one. In the following, the value obtained by the resolution of the discretized system is $\psi_\mathrm{P} = \psi(\mathrm{P}) + O(\varepsilon(\mathbf{dx}, dt))$, with $O(\varepsilon(\mathbf{dx}, dt))$ the discretization error depending on the chosen method.

The same approximation can be done considering a face f and its geometric center F:

$$\iint_{S_\mathrm{f}} \psi\, dS \approx S_\mathrm{f}\, \psi(\mathrm{F}) \tag{3.2}$$

where $\psi(\mathrm{F})$ is the value of ψ at F.

3.1.3 Cell to Face center interpolation

Locally, the fluid velocity allows to define an upwind and a downwind direction. To introduce the face interpolation schemes properly, it is assumed here as a convention that the point N is located downwind the point P. The upwind point U is located such that $\mathbf{UP} = \mathbf{PN}$ (Fig. 3.1).

Given a volumic field ψ, a f and face separating a cell P and a cell N, evaluating $\psi(\mathrm{F})$ is done using a face interpolation scheme. These interpolations are especially used for the discretisation of the advection term (see Sec. 3.1.5).

The face interpolation is done using the High Resolution Schemes (HRS). In the context of unstructured three-dimensional meshes as the ones used in OpenFOAM, a typical HRS formulation is the "inverse-distance weighting flux-limiter" formulation as proposed by Hou et al. (2012). In this formulation, the approximated value ψ_f of the quantity $\psi(\mathrm{F})$ is defined as:

$$\psi(\mathrm{F}) \approx \psi_\mathrm{f} = \psi_\mathrm{P} + \frac{\Xi(r)}{L}\,(\psi_\mathrm{N} - \psi_\mathrm{P}) \tag{3.3}$$

where L is the weight defined as

$$L := \frac{|\mathbf{PF} \bullet \mathbf{n}_\mathrm{f}| + |\mathbf{NF} \bullet \mathbf{n}_\mathrm{f}|}{|\mathbf{PF} \bullet \mathbf{n}_\mathrm{f}|} \tag{3.4}$$

$\Xi(r)$ is the so-called flux-limiter function of r the gradient ratio defined as:

$$r := \frac{\psi_\mathrm{P} - \psi_\mathrm{U}}{\psi_\mathrm{N} - \psi_\mathrm{P}} \tag{3.5}$$

The value of ψ_U is the virtual value of the virtual upwind point U (Fig. 3.1). On unstructured meshes, the value of ψ_U can be approximated using the second-order method proposed by Darwish and Moukalled (2003). This method is based on the evaluation of the gradient of ψ at P,

$$\nabla(\psi)(\mathrm{P}) \approx \nabla\psi|_\mathrm{P} \tag{3.6}$$

This results in the following definition:

$$\psi_\mathrm{U} = \psi_\mathrm{N} - 2\,\nabla\psi|_\mathrm{P} \bullet \mathbf{PN} \tag{3.7}$$

and then:

$$r = \frac{2\,\nabla\psi|_\mathrm{P} \bullet \mathbf{PN}}{\psi_\mathrm{N} - \psi_\mathrm{P}} - 1 \tag{3.8}$$

which is the formulation used in the solver. In this approximation, the quantity $\nabla\psi|_\mathrm{P}$ has to be explicitly expressed.

The discretisation scheme depends on the choice of the limiter in Eq. 3.3.

a) Convection Boundedness Criterion

In order to guarantee the boundedness of a solution of advective terms using a discretization scheme Gaskell and Lau (1988) formulated the Convection Boundedness Criterion (CBC). First, it is useful to introduce the normalized variable $\hat{\psi} = \frac{\psi - \psi_\mathrm{U}}{\psi_\mathrm{N} - \psi_\mathrm{U}}$. Then, the CBC assumes that a normalized variable $\hat{\psi}_\mathrm{f}$ (a numerical approximation of $\hat{\psi}(\mathrm{F})$) is bounded if:

$$\hat{\psi}_P < \hat{\psi}_f < 1 \text{ if } 0 < \hat{\psi}_P < 1 \tag{3.9a}$$

$$\hat{\psi}_f = \hat{\psi}_P \text{ if } \hat{\psi}_P < 0 \text{ or } \hat{\psi}_P > 1 \tag{3.9b}$$

$$\hat{\psi}_f = 0 \text{ if } \hat{\psi}_P \leq 0 \tag{3.9c}$$

$$\hat{\psi}_f = 1 \text{ if } \hat{\psi}_P \geq 1 \tag{3.9d}$$

Using the flux-limiter formulation (Eq. 3.3), a sufficient condition for the CBC is:

$$0 < \Xi(r) \leq L \text{ if } r > 0 \tag{3.10a}$$

$$\Xi(r) = 0 \text{ if } r \leq 0 \tag{3.10b}$$

$$\tag{3.10c}$$

b) Total variation diminishing criterium

The Total Variation Diminishing (TVD) criterion is based on the articles of Harten (1984) and Sweby (1984). The objective was, first, to prevent the solution from unstable oscillations that can occur using some second-order schemes, and then, to preserve the monotonicity [1]. Consequently, the TVD criterion prevents the creation of new local extrema. Using the Sweby (1984) flux-limiter formulation combined with the Hou et al. (2012) inverse-distance weighting, the TVD criterion results is formulated as follows:

$$0 < \Xi(r) \leq Lr \text{ if } 0 < r < 1 \tag{3.11a}$$

$$0 < \Xi(r) \leq L \text{ if } r \geq 1 \tag{3.11b}$$

$$\Xi(r) = 0 \text{ if } r \leq 0 \tag{3.11c}$$

It can be noted that these criteria on flux-limiter encompass the CBC. Furthermore, the following additional conditions ensure second-order accuracy of the schemes for regular meshes:

1. a monotonic function in space at time t remains monotonic at time $t + \Delta t$

$$r \leq \Xi(r) \leq Lr \text{ if } 0 < r \leq 1/2 \tag{3.12a}$$

$$r \leq \Xi(r) \leq 1 \text{ if } 1/2 \leq r \leq 1 \tag{3.12b}$$

$$1 \leq \Xi(r) \leq r \text{ if } 1 \leq r \leq L \tag{3.12c}$$

$$1 \leq \Xi(r) \leq L \text{ if } r \geq L \tag{3.12d}$$

$$\Xi(r) = 0 \text{ if } r \leq 0 \tag{3.12e}$$

These conditions can be illustrated on the Sweby (1984) diagram (Fig. 3.2) where the TVD region is shaded in gray. Light gray represents the first-order regions and dark gray the second-order regions.

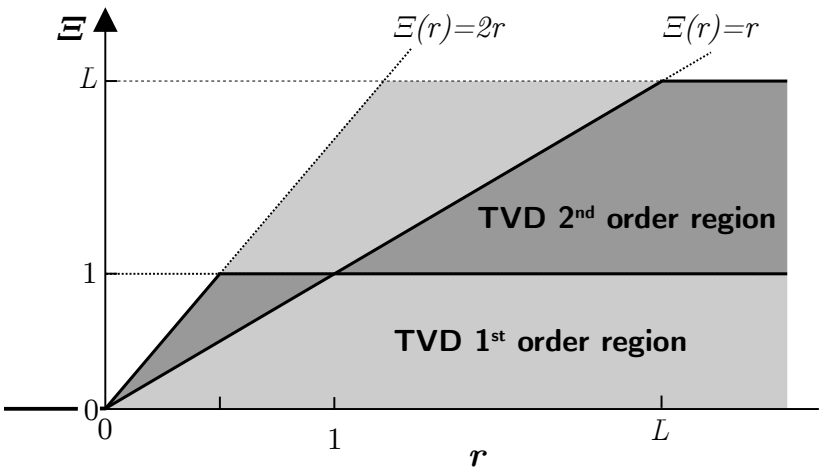


Figure 3.2 – Flux limiter Ξ as a function of r with TVD first-order and second-order regions

c) Usual interpolation schemes

Upwind differencing scheme (UD) With $\Xi(r) = 0$ the resulting scheme is the upwind differencing:

$$\psi_f = \psi_P \tag{3.13}$$

This scheme is unconditionally stable for the resolution of convective terms but it is a formally a first-order accurate scheme.

Central differencing scheme (CD) With $\Xi(r) = 1$ the resulting scheme is the central differencing:

$$\psi_{\mathrm{f}} = \left(1 - \frac{1}{L}\right)\psi_{\mathrm{P}} + \frac{\psi_{\mathrm{N}}}{L} \tag{3.14}$$

For meshes where the points P, F and N are aligned, this CD scheme is formally second-order accurate but can lead to instabilities in the implicit resolution of convective terms.

d) Mesh skewness

Unstructured meshes do not necessarily respect the property of P, F and N alignment. Figure 3.1 is an example of such a non-alignment. This non-alignment is often referred to as the skewness (Jasak, 1996). The point F' in Figure 3.1 is the intersection of the segment PN and face f. The vector **F'F** is the skewness correction vector of the face f.

As mentioned, the face integral approximation (Eq. 3.2) reaches a second-order accuracy if ψ_{f} is a second-order approximation of $\psi(\mathrm{F})$. However, with skewed unstructured meshes, the CD scheme gives formally a second-order approximation of $\psi(F')$ but not of $\psi(\mathrm{F})$. Consequently, with skewed faces, using Eq. 3.14 for ψ_{f} formulation comes to consider Eq. 3.2 with the first-order approximation that $\psi(\mathrm{F}) \approx \psi(F')$. Finally, with skewed cells, the following approximation:

$$\iint_{S_{\mathrm{f}}} \psi \, dS \approx S_{\mathrm{f}} \left[\left(1 - \frac{1}{L}\right)\psi_{\mathrm{P}} + \frac{\psi_{\mathrm{N}}}{L}\right] \tag{3.15}$$

is formally a first-order approximation.

The given example with CD scheme can be extended to any other HRS formulated with Eq. 3.3 evaluation and then, without using corrector terms, the skewness of a mesh reduces the face integral approximation to first-order accuracy.

In order to keep a second-order formulation a skew-corrector can be formulated. To do so, the gradient of ψ has to be evaluated at P with:

$$\nabla(\psi)(\mathrm{F}) \approx \nabla\psi|_{\mathrm{f}} \tag{3.16}$$

The resulting second-order approximation is:

$$\psi(\mathrm{F}) \approx \psi(\mathrm{F'}) + \nabla\psi|_{\mathrm{f}} \bullet \mathbf{F'F} \tag{3.17}$$

Finally, with a skew-corrected CD scheme for instance, the following second-order

approximation can be made:

$$\iint_{S_\mathrm{f}} \psi \, dS \approx S_\mathrm{f} \left[\left(1 - \frac{1}{L} \right) \psi_\mathrm{P} + \frac{\psi_\mathrm{N}}{L} + \underbrace{\nabla \psi|_\mathrm{f} \bullet \mathbf{F'F}}_{\textit{skew-corrector}} \right] \tag{3.18}$$

In this formulation the $\nabla \psi|_\mathrm{f}$ has to be known before solving the equation.

3.1.4 Face to Cell center reconstruction

For a vector field $\boldsymbol{\psi}$, with $\boldsymbol{\psi}_\mathrm{f}$ the quantity $\boldsymbol{\psi}(\mathrm{F})$ evaluated at the face center, a second order approximation of the cell-center value $\boldsymbol{\psi}_\mathrm{P}$ can be constructed using the method proposed by Weller and Shahrokhi (2014):

$$\boldsymbol{\psi}_\mathrm{P} = \left[\sum_\mathrm{f} \mathbf{n}_\mathrm{f} \otimes \mathbf{S}_\mathrm{f} \right]^{-1} \sum_\mathrm{f} (\boldsymbol{\psi}_\mathrm{f} \bullet \mathbf{n}_\mathrm{f}) \, \mathbf{S}_\mathrm{f} \tag{3.19}$$

Thanks to this equation, a second-order reconstruction of any vector field at the geometric cell center can be done using the surface normal quantities $\boldsymbol{\psi}_\mathrm{f} \bullet \mathbf{n}_\mathrm{f}$.

3.1.5 Divergence

As mentioned, the fundamental aspect of the Finite Volume method is to consider elementary volumes rather than local nodes. Consequently, for a vector field $\boldsymbol{\psi}$, the Gauss theorem is applied in order to transform the volumic integration over a cell of $\nabla \bullet \boldsymbol{\psi}$ into a surface integration as follows:

$$\iiint_{V_\mathrm{P}} \nabla \bullet \boldsymbol{\psi} dV = \sum_\mathrm{f} \iint_S \boldsymbol{\psi} \bullet \boldsymbol{dS} \tag{3.20}$$

Then, using the cell and face-center approximation results in the second-order approximation:

$$\nabla \bullet \boldsymbol{\psi}|_\mathrm{P} \approx \frac{1}{V_\mathrm{P}} \sum_\mathrm{f} \boldsymbol{\psi}_\mathrm{f} \bullet \boldsymbol{S}_\mathrm{f} \tag{3.21}$$

With this formulation of the divergence, when the field is a velocity field $\mathbf{u}$ it is convenient to introduce a dedicated notation of the flux ϕ_f such that for a face f

$$\phi_\mathrm{f} = \mathbf{u}_\mathrm{f} \bullet \mathbf{S}_\mathrm{f} \approx \iint_f \mathbf{u} \bullet d\boldsymbol{S}, \tag{3.22}$$

and then

$$\nabla \bullet \mathbf{u}|_\mathrm{P} \approx \frac{1}{V_\mathrm{P}} \sum_\mathrm{f} \phi_\mathrm{f} \tag{3.23}$$

3.1.6 Surface normal gradient

The Figure 3.3 shows a cell face from a *non-orthogonal mesh*. In OpenFOAM, a mesh is said to be *orthogonal* if and only if the line joining the centers of neighboring cells (here P and N) and the shared face f are orthogonal. Figure 3.3 illustrates the following decomposition:

$$\boldsymbol{S}_\mathrm{f} = \boldsymbol{\Delta} + \boldsymbol{k} \tag{3.24}$$

Different expressions of $\boldsymbol{\Delta}$ can be chosen. The one used in the presented work is named "over relaxed approach" in Jasak (1996) and is defined as follows:

$$\boldsymbol{\Delta} = \frac{|\mathbf{S_f}|^2}{\mathbf{PN} \bullet \boldsymbol{S}_\mathrm{f}} \mathbf{PN} \tag{3.25}$$

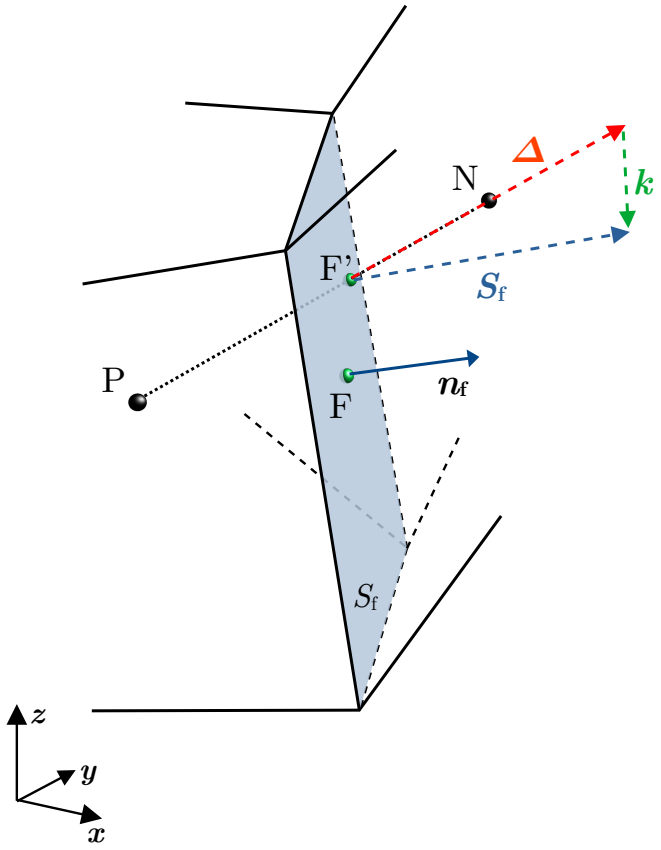


Figure 3.3 – A cell face of a non-orthogonal mesh

Using this decomposition of $\boldsymbol{S}_\mathrm{f}$, the quantity $\nabla\psi|_\mathrm{f} \cdot \boldsymbol{S}_\mathrm{f}$ for a field ψ can be decomposed as a *orthogonal part* and an *non-orthogonal part*:

$$\nabla\psi|_\mathrm{f} \cdot \boldsymbol{S}_\mathrm{f} = \underbrace{\nabla\psi|_\mathrm{f} \cdot \boldsymbol{\Delta}}_{orthogonal\ part} + \underbrace{\nabla\psi|_\mathrm{f} \cdot \boldsymbol{k}}_{non-orthogonal\ part} \tag{3.26}$$

Using the Eq. 3.25 and the second-order approximation:

$$\nabla\psi \cdot \mathbf{PN} \approx \psi_\mathrm{N} - \psi_\mathrm{P} \tag{3.27}$$

the following approximation can be done:

$$\nabla\psi|_{\mathrm{f}} \cdot \boldsymbol{S}_{\mathrm{f}} \approx \underbrace{\frac{|\boldsymbol{S}_{\mathrm{f}}|^2}{\mathbf{PN} \cdot \boldsymbol{S}_{\mathrm{f}}}(\psi_{\mathrm{N}} - \psi_{\mathrm{P}})}_{orthogonal\ part} + \underbrace{\nabla\psi|_{\mathrm{f}} \cdot (\boldsymbol{S}_{\mathrm{f}} - \boldsymbol{\Delta})}_{non-orthogonal\ part} \tag{3.28}$$

3.1.7 Non-orthogonal correction

During the resolution of the discretized system, the use of Eq. 3.28 to evaluate $\nabla\psi|_{\mathrm{f}} \cdot \boldsymbol{S}_{\mathrm{f}}$ implies the use of an explicitly known approximation of $\nabla\psi|_{\mathrm{f}}$ in the right-hand member. Similarly to the skew-correction seen in Section 3.1.3, because of its explicit aspect, the *non-orthogonal part* of the Eq. 3.28 will be considered as a *non-orthogonal-corrector*. Some other discretization schemes that would not imposed the use of explicit terms could have been chosen but as explained by Jasak (1996), the use of this scheme is justified by a reduced size of the discretization molecule and a reduced error.

3.1.8 Cell center gradient

For a scalar field ψ the value of the cell center gradient $\nabla\psi|_{\mathrm{P}}$ can be approximated using several methods.

Gauss linear

$$\iiint_{V_{\mathrm{P}}} \nabla\psi dV = \sum_{\mathrm{f}} \iint_{S} \psi d\boldsymbol{S} \tag{3.29}$$

then, using the cell and face-center approximations:

$$\nabla\psi|_{\mathrm{P}} \approx \frac{1}{V_{\mathrm{P}}} \sum_{\mathrm{f}} \psi_{\mathrm{f}} \boldsymbol{S}_{\mathrm{f}} \tag{3.30}$$

Least Square This method consists in minimizing the square of the weighted error defined as:

$$e^2 = \sum_{\mathrm{N}} \left| \frac{\psi_{\mathrm{N}} - \psi_{\mathrm{P}}}{|\mathbf{PN}|} - \frac{\mathbf{PN}}{|\mathbf{PN}|} \cdot \nabla\psi|_{\mathrm{P}} \right| \tag{3.31}$$

The resulting solution is:

$$\nabla\psi|_{\mathrm{P}} \approx \sum_{\mathrm{N}} \mathbf{G}^{-1} \cdot \frac{\mathbf{PN}}{|\mathbf{PN}|^2}(\psi_{\mathrm{N}} - \psi_{\mathrm{P}}) \tag{3.32}$$

with:

$$\mathbf{G} = \sum_{\mathrm{N}} \frac{\mathbf{PN} \otimes \mathbf{PN}}{|\mathbf{PN}|^2} \tag{3.33}$$

The interest of this scheme is the minimisation of the error.

3.2 Time discretisation

The equations with time derivative terms of a field ψ can be presented under the following form:

$$\left.\frac{\partial \psi}{\partial t}\right|_t = \mathscr{G}(\psi, \mathbf{x}, t) \tag{3.34}$$

where $\mathscr{G}$ is a function of ψ, $\mathbf{x}$ and t, free of time derivative terms.

Then the discretization of Eq. 3.34 can be written as follows:

$$\frac{\partial \psi^{(\mathrm{n})}}{\partial t} = c_{\mathrm{n}}^{(\mathrm{n})} \psi^{(\mathrm{n})} + \underbrace{c_{\mathrm{n\text{-}1}}^{(\mathrm{n})} \psi^{(\mathrm{n\text{-}1})} + c_{\mathrm{n\text{-}2}}^{(\mathrm{n})} \psi^{(\mathrm{n\text{-}2})} + \gamma_{\psi}^{(\mathrm{n})}}_{S_{\mathrm{ddt}\psi}^{(\mathrm{n})}} \tag{3.35}$$

The term $\psi^{(\mathrm{n})}$ identifies the computed value of ψ at the n^{th} time-step. The coefficients $c_{\mathrm{n}}^{(\mathrm{n})}$, $c_{\mathrm{n\text{-}1}}^{(\mathrm{n})}$, $c_{\mathrm{n\text{-}2}}^{(\mathrm{n})}$ depend on the discretisation scheme used at the n^{th} time-step. $\gamma_{\psi}^{(\mathrm{n})}$ is an additional explicit term depending on both the selected time schemes and the variable ψ. The order of accuracy of the time approximation (Eq. 3.35) depends on the choice of the coefficients $c_{\mathrm{n}}^{(\mathrm{n})}$, $c_{\mathrm{n\text{-}1}}^{(\mathrm{n})}$, $c_{\mathrm{n\text{-}2}}^{(\mathrm{n})}$ and γ_{ψ}. For ease the explicit part of this formulation can occasionally be grouped as a temporal discretization source term $S_{\mathrm{ddt}\psi}^{(\mathrm{n})}$:

$$S_{\mathrm{ddt}\psi}^{(\mathrm{n})} := c_{\mathrm{n\text{-}1}}^{(\mathrm{n})} \psi^{(\mathrm{n\text{-}1})} + c_{\mathrm{n\text{-}2}}^{(\mathrm{n})} \psi^{(\mathrm{n\text{-}2})} + \gamma_{\psi}^{(\mathrm{n})} \tag{3.36}$$

Eq. 3.35 and Eq. 3.36 are the general form of any temporal discretization scheme that are used in this document.

3.2.1 Implicit Euler and second-order Backward schemes

a) Implicit Euler

Using the Implicit Euler scheme, Eq. 3.34 can be discretized as follows:

$$\frac{\psi^{(\mathrm{n})} - \psi^{(\mathrm{n\text{-}1})}}{\Delta t^{(\mathrm{n})}} = \mathscr{G}^{(\mathrm{n})} \tag{3.37}$$

With $\mathscr{G}^{(\mathrm{n})}$ the numerical approximation of $\mathscr{G}\left(t_{(n)}\right)$, and $\Delta t^{(\mathrm{n})} := t(n) - t(n-1)$ the time interval between the (n-1)$^{\text{th}}$ and n$^{\text{th}}$ time step. This time step size is not necessarily constant all along the simulation. Then, the following identification can be done:

$$c_{\mathrm{n}}^{(\mathrm{n})} := \frac{1}{\Delta t^{(\mathrm{n})}} \tag{3.38a}$$

$$c_{\mathrm{n\text{-}1}}^{(\mathrm{n})} := -\frac{1}{\Delta t^{(\mathrm{n})}} \tag{3.38b}$$

$$c_{\mathrm{n\text{-}2}}^{(\mathrm{n})} := 0 \tag{3.38c}$$

$$\gamma_{\psi}^{(\mathrm{n})} := 0 \tag{3.38d}$$

This numerical scheme is non-conditionally stable and is first-order accurate. Consequently, it will generate significant numerical damping.

b) Second-order Backward

Using the Second-order Backward scheme, Eq. 3.34 can be discretized as follows:

$$\frac{\Delta t^{(\mathrm{n\text{-}1})}(2\Delta t^{(\mathrm{n})} + \Delta t^{(\mathrm{n\text{-}1})})\,\psi^{(\mathrm{n})} - (\Delta t^{(\mathrm{n})} + \Delta t^{(\mathrm{n\text{-}1})})^2\,\psi^{(\mathrm{n\text{-}1})} + {\Delta t^{(\mathrm{n})}}^2\,\psi^{(\mathrm{n\text{-}2})}}{\Delta t^{(\mathrm{n\text{-}1})}(\Delta t^{(\mathrm{n})} + \Delta t^{(\mathrm{n\text{-}1})})} = \mathscr{G}^{(\mathrm{n})} \tag{3.39}$$

with

$$\Delta t^{(\mathrm{n})} := t(n) - t(n-1) \tag{3.40a}$$

$$\Delta t^{(\mathrm{n\text{-}1})} := t(n-1) - t(n-2) \tag{3.40b}$$

Using the Second-order Backward scheme, the following identification can be done:

$$c_{\mathrm{n}}^{(\mathrm{n})} := \frac{1}{\Delta t^{(\mathrm{n})}} + \frac{1}{\Delta t^{(\mathrm{n})} + \Delta t^{(\mathrm{n\text{-}1})}} \tag{3.41a}$$

$$c_{\mathrm{n\text{-}1}}^{(\mathrm{n})} := -\frac{\Delta t^{(\mathrm{n})} + \Delta t^{(\mathrm{n\text{-}1})}}{\Delta t^{(\mathrm{n})} \Delta t^{(\mathrm{n\text{-}1})}} \tag{3.41b}$$

$$c_{\mathrm{n\text{-}2}}^{(\mathrm{n})} := \frac{\Delta t^{(\mathrm{n})}}{\Delta t^{(\mathrm{n\text{-}1})} \left(\Delta t^{(\mathrm{n})} + \Delta t^{(\mathrm{n\text{-}1})}\right)} \tag{3.41c}$$

$$\gamma_{\psi}^{(\mathrm{n})} := 0 \tag{3.41d}$$

This scheme is a second-order accuracy one and is unconditionally stable.

3.2.2 Crank-Nicolson schemes

The Crank-Nicolson method is the name given to a second-order approximation of Eq. 3.34:

$$\frac{\psi^{(\mathrm{n})} - \psi^{(\mathrm{n\text{-}1})}}{\Delta t^{(\mathrm{n})}} = \frac{1}{2}\left[\mathscr{G}^{(\mathrm{n})} + \mathscr{G}^{(\mathrm{n\text{-}1})}\right], \tag{3.42}$$

In the literature, it is said that this trapezoïd rule method applied to Navier-Stokes equations can lead to oscillatory behavior during the computation. Those instabilities can be reduced by introducing a blending factor $C_{\mathrm{CN}} \in [0.5, 1]$ to Eq. 3.42. It results in the following formulation:

$$\frac{\psi^{(\mathrm{n})} - \psi^{(\mathrm{n\text{-}1})}}{\Delta t^{(\mathrm{n})}} = C_{\mathrm{CN}} \mathscr{G}^{(\mathrm{n})} + (1 - C_{\mathrm{CN}}) \mathscr{G}^{(\mathrm{n\text{-}1})} \tag{3.43}$$

This equation can be rearranged as follow:

$$\frac{\psi^{(\mathrm{n})} - \psi^{(\mathrm{n\text{-}1})}}{C_{\mathrm{CN}} \Delta t^{(\mathrm{n})}} - \frac{1 - C_{\mathrm{CN}}}{C_{\mathrm{CN}}} \mathscr{G}^{(\mathrm{n\text{-}1})} = \mathscr{G}^{(\mathrm{n})} \tag{3.44}$$

Finally, considering Eq. 3.43, Def. 3.35 and Def. 3.36, the following identification can be done:

$$c_{\mathrm{n}}^{(\mathrm{n})} := \frac{1}{C_{\mathrm{CN}}\Delta t^{(\mathrm{n})}} \tag{3.45a}$$

$$c_{\mathrm{n\text{-}1}}^{(\mathrm{n})} := -\frac{1}{C_{\mathrm{CN}}\Delta t^{(\mathrm{n})}} \tag{3.45b}$$

$$c_{\mathrm{n\text{-}2}}^{(\mathrm{n})} := 0 \tag{3.45c}$$

$$\gamma_{\psi}^{(\mathrm{n})} := -\frac{1-C_{\mathrm{CN}}}{C_{\mathrm{CN}}}\mathscr{G}^{(\mathrm{n\text{-}1})} \tag{3.45d}$$

The blending factor C_{CN} controls the weight of the implicit term $\mathscr{G}^{(\mathrm{n})}$ with respect to the explicit term $\mathscr{G}^{(\mathrm{n\text{-}1})}$ in Eq. 3.42. This Crank-Nicolson formulation is formally second-order accurate only if $C_{\mathrm{CN}} = 0.5$. With $C_{\mathrm{CN}} = 1$, Eq. 3.42 is equivalent to the implicit Euler method.

OpenFOAM formulation In the discretization process of OpenFOAM (except for the VOF equation) the term $\mathscr{G}^{(\mathrm{n\text{-}1})}$ is generally difficult to estimate in the time discretization algorithm. An alternative is used by getting access to the values of $\psi^{(\mathrm{n\text{-}1})}$ and $\psi^{(\mathrm{n\text{-}2})}$. This is obtained applying Eq. 3.44 at the time-step $(n-1)$:

$$\frac{\psi^{(\mathrm{n\text{-}1})} - \psi^{(\mathrm{n\text{-}2})}}{C_{\mathrm{CN}}\Delta t^{(\mathrm{n\text{-}1})}} - \frac{1-C_{\mathrm{CN}}}{C_{\mathrm{CN}}}\mathscr{G}^{(\mathrm{n\text{-}2})} = \mathscr{G}^{(\mathrm{n\text{-}1})} \tag{3.46}$$

and identifying the recursive relation:

$$\gamma^{(\mathrm{n})} := -\frac{1-C_{\mathrm{CN}}}{C_{\mathrm{CN}}}\left[\frac{\psi^{(\mathrm{n\text{-}1})} - \psi^{(\mathrm{n\text{-}2})}}{C_{\mathrm{CN}}\Delta t^{(\mathrm{n\text{-}1})}} + \gamma^{(\mathrm{n\text{-}1})}\right] \tag{3.47}$$

For more clarity, a so-called Crank-Nicolson off-centre coefficient $C_{\mathrm{OC}} \in [0,1]$ is defined as:

$$C_{\mathrm{OC}} := \frac{1-C_{\mathrm{CN}}}{C_{\mathrm{CN}}} \tag{3.48}$$

This results in the following definition:

$$c_{\mathrm{n}}^{(\mathrm{n})} := \frac{1+C_{\mathrm{OC}}}{\Delta t^{(\mathrm{n})}} \tag{3.49a}$$

$$c_{\mathrm{n\text{-}1}}^{(\mathrm{n})} := -\frac{1+C_{\mathrm{OC}}}{\Delta t^{(\mathrm{n})}} \tag{3.49b}$$

$$c_{\mathrm{n\text{-}2}}^{(\mathrm{n})} := 0 \tag{3.49c}$$

$$\gamma_{\psi}^{(\mathrm{n})} := -C_{\mathrm{OC}}\left[\frac{1+C_{\mathrm{OC}}}{\Delta t^{(\mathrm{n\text{-}1})}}\left(\psi^{(\mathrm{n\text{-}1})}-\psi^{(\mathrm{n\text{-}2})}\right)+\gamma^{(\mathrm{n\text{-}1})}\right]. \tag{3.49d}$$

using the finite-volume method.

3.2.3 Finite-volume method and moving meshes

The Navier-Stokes equations with or without turbulence model are unsteady and comprise a first-order time derivative. In the domain reference-frame $\mathscr{R}_0$, considering a generic field ψ, using Finite Volume Method (FVM), the following integral form has to be approximated:

$$\iiint_{V_{\mathrm{P}}} \left.\frac{\partial \psi}{\partial t}\right|_{\mathscr{R}_0} dV$$

To take into account a possible mesh motion, this term becomes:

$$\iiint_{V_{\mathrm{P}}} \left.\frac{\partial \psi}{\partial t}\right|_{\mathscr{R}_0} dV = \frac{\mathrm{d}}{\mathrm{d}t}\left(\iiint_{V_{\mathrm{P}}(t)} \psi\, dV\right)_{\mathscr{R}_0} - \sum_{\mathrm{f}(t)} \iint_{S_{\mathrm{f}}} \psi\, \mathbf{v}_{\mathrm{m}} \bullet \mathbf{dS} \tag{3.50}$$

where $\mathbf{v}_{\mathrm{m}}$ represents the velocity of the elementary surface of integration on S_{f} in $\mathscr{R}_0$. This velocity is induced by the mesh motion. Besides, using the spatial discretization mentioned previously, the following approximation can be done:

$$\left.\frac{\partial \psi_{\mathrm{P}}}{\partial t}\right|_{\mathscr{R}_0}(t^{(\mathrm{n})}) \approx \left.\frac{\partial \psi_{\mathrm{P}}}{\partial t}\right|_{\mathscr{R}_0}^{(\mathrm{n})} = \frac{1}{V_{\mathrm{P}}}\left(\left(\frac{\mathrm{d}\psi_{\mathrm{P}} V_{\mathrm{P}}}{\mathrm{d}t}\right)^{(\mathrm{n})} - \sum_{\mathrm{f}} \psi_{\mathrm{f}}^{(\mathrm{n})}\, \phi_{\mathrm{mf}}^{(\mathrm{n})}\right) \tag{3.51}$$

where ϕ_{mf} is the mesh velocity flux at the face f. Then, using the previously defined formulation for the time discretization schemes (Eq. 3.35):

$$\left.\frac{\partial \psi_{\mathrm{P}}}{\partial t}\right|_{\mathscr{R}_0}^{(\mathrm{n})} = c_{\mathrm{n}}^{(\mathrm{n})}\psi^{(\mathrm{n})} + c_{\mathrm{n\text{-}1}}^{(\mathrm{n})}\psi^{(\mathrm{n\text{-}1})}\frac{V_{\mathrm{P}}^{(\mathrm{n\text{-}1})}}{V_{\mathrm{P}}^{(\mathrm{n})}} + c_{\mathrm{n\text{-}2}}^{(\mathrm{n})}\psi^{(\mathrm{n\text{-}2})}\frac{V_{\mathrm{P}}^{(\mathrm{n\text{-}2})}}{V_{\mathrm{P}}^{(\mathrm{n})}} + \frac{\gamma_{\psi V_{\mathrm{P}}}^{(\mathrm{n})}}{V_{\mathrm{P}}^{(\mathrm{n})}} - \frac{1}{V_{\mathrm{P}}^{(\mathrm{n})}}\sum_{\mathrm{f}} \psi_{\mathrm{f}}^{(\mathrm{n})}\, \phi_{\mathrm{mf}}^{(\mathrm{n})} \tag{3.52}$$

Given a constant field ψ in space and time, Eq. 3.51 implies the following conservation law:

$$\left(\frac{\mathrm{d}V_\mathrm{P}}{\mathrm{d}t}\right)^{(\mathrm{n})} = \sum_\mathrm{f} \phi_\mathrm{mf}^{(\mathrm{n})} \tag{3.53}$$

and using the coefficients of the selected time-scheme:

$$c_\mathrm{n}^{(\mathrm{n})} V_\mathrm{P}^{(\mathrm{n})} + c_\mathrm{n\text{-}1}^{(\mathrm{n})} V_\mathrm{P}^{(\mathrm{n\text{-}1})} + c_\mathrm{n\text{-}2}^{(\mathrm{n})} V_\mathrm{P}^{(\mathrm{n\text{-}2})} + \gamma_{V_\mathrm{P}}^{(\mathrm{n})} = \sum_\mathrm{f} \phi_\mathrm{mf}^{(\mathrm{n})} \tag{3.54}$$

Consequently, respecting the conservation law, $\phi_\mathrm{mf}^{(\mathrm{n})}$ formulation depends on the chosen time scheme.

3.3 Solver algorithm

The *foamStar* algorithm is detailed in this section. For simulations with bodies, only rigid body motions are considered here [2]. The physical system of equations solved by *foamStar* is the one established in Section 2.1. Table 3.1 indicates the modeled quantities to be solved and the set of equations is recalled in the following.

Table 3.1 – Modeled quantities evaluated with VOF-RANSE resolution

Physical quantity	**Description**	**Unit**	***foamStar* notation**
$\mathbf{u}$	Fluid velocity	[m.s^{-1}]	`U`
p	Pressure	[Pa]	`p_rgh`
α	Volume-fraction	[ ]	`alpha`
ρ	Volumetric mass density	[kg.m^{-3}]	`rho`
ν	Kinematic viscosity	[m^2.s^{-1}]	`nu`
ν_t	Turbulent kinematic viscosity	[m^2.s^{-1}]	`nut`
k	Turbulent kinetic energy	[m^2.s^{-2}]	`k`
ω	Turbulent dissipation rate	[s^{-1}]	`omega`

VOF The VOF convection equation is:

$$\frac{\partial \alpha}{\partial t} + \nabla\bullet(\alpha\mathbf{u}) + \nabla\bullet\mathbf{u}_\mathrm{comp} = 0 \tag{2.18}$$

2. An hydroelastic model is also implemented in *foamStar* (Seng et al., 2014)

and the mixture equations are:

$$\rho = \alpha\rho_{\mathrm{w}} + (1-\alpha)\rho_{\mathrm{a}} \tag{2.5a}$$

$$\mu = \alpha\mu_{\mathrm{w}} + (1-\alpha)\mu_{\mathrm{a}} \tag{2.5b}$$

$$\nu = \frac{\mu}{\rho} \tag{2.5c}$$

RANSE The internal flow model is based on RANSE with the following continuity equation:

$$\nabla \bullet \mathbf{u} = 0 \tag{2.7}$$

and momentum equation:

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla\bullet(\rho\mathbf{u}\mathbf{u} + (\rho_{\mathrm{w}} - \rho_{\mathrm{a}})\mathbf{u}_{\mathrm{comp}}\mathbf{u}) - \nabla p = -\mathbf{g}\bullet\mathbf{x}\nabla\rho + \nabla\bullet(\mu_{\mathrm{eff}}\nabla\mathbf{u}) + \nabla\mu_{\mathrm{eff}}\bullet\nabla\mathbf{u}^T - \rho\,\mathbf{a}_{\mathrm{cc}_0} \tag{2.29}$$

with

$$\mu_{\mathrm{eff}} = \mu + \rho\nu_{\mathrm{t}} \tag{2.21}$$

and

$$p = p_{\mathrm{d}} + \frac{2}{3}\rho\,k \tag{2.22}$$

Turbulent Model (TM) The two-equation fs-k-ω-SST model is:

$$\frac{\partial\rho k}{\partial t} + \nabla\bullet\rho\mathbf{u}k - \nabla\bullet((\mu + \rho\alpha_k\nu_{\mathrm{t}})\boldsymbol{\nabla}k) = \rho\nu_{\mathrm{t}}p_0 - \frac{2}{3}\rho k\nabla\bullet\mathbf{u} - \rho\beta^* k\omega - \rho P_{\mathrm{b}} \tag{2.24a}$$

$$\frac{\partial\rho\omega}{\partial t} + \nabla\bullet\rho\mathbf{u}\omega - \nabla\bullet((\mu + \rho\alpha_\omega\nu_{\mathrm{t}})\boldsymbol{\nabla}\omega) = \gamma\rho p_0 - \frac{2}{3}\rho\gamma\omega\nabla\bullet\mathbf{u} - \rho\beta^*\omega^2 - \rho(F_1 - 1)CD_{k\omega} \tag{2.24b}$$

FSI For a rigid body, the equations given by the FSI are:

$$\overline{\overline{\Sigma}} = -\left[p + \rho \mathbf{g} \bullet \mathbf{x}\right] \overline{\overline{\mathrm{I}}} + \mu_{\mathrm{eff}} \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right) \tag{2.44}$$

$$\mathbf{F}_{\mathrm{fluid}} = \int_{\delta\Omega_{body}} \mathbf{dS} \bullet \overline{\overline{\Sigma}} \tag{2.43a}$$

$$\mathbf{M}^{\mathrm{b}}_{\mathrm{fluid}} = \mathbf{R}^{\mathrm{b}}_{0} \left[\int_{M \in \delta\Omega_{body}} \mathbf{dS} \bullet \overline{\overline{\Sigma}} \wedge \mathbf{C_g M} \right] \tag{2.43b}$$

$$m \ddot{\mathbf{X}}_{\mathrm{Cg}} = m\mathbf{g} + \mathbf{F}_{\mathrm{fluid}} + \mathbf{F}_{\mathrm{add}} + \mathbf{F}_{\mathrm{c}} \tag{2.41}$$

$$\mathbf{I}_{\mathrm{g}} \dot{\boldsymbol{\omega}}^{\mathrm{b}} = \mathbf{M}^{\mathrm{b}}_{\mathrm{fluid}} + \mathbf{M}^{\mathrm{b}}_{\mathrm{add}} + \boldsymbol{\omega}^{\mathrm{b}} \wedge \left(\mathbf{I}_{\mathrm{g}} \boldsymbol{\omega}^{\mathrm{b}} \right) \tag{2.42}$$

$$\forall M \in \partial\Omega_{body}, \quad \mathbf{X}_{\mathrm{M}} = \mathbf{X}_{\mathrm{Cg}} + \mathbf{R}^{0}_{\mathrm{b}} \left(\mathbf{C_g M}^{\mathrm{b}} \right) \tag{2.45a}$$

$$\mathbf{u}(M) = \mathbf{u}_{\mathrm{wall}}(M) = \dot{\mathbf{X}}_{\mathrm{Cg}} + \mathbf{R}^{0}_{\mathrm{b}} \left[\mathbf{C_g M}^{\mathrm{b}} \wedge \boldsymbol{\omega}^{\mathrm{b}} \right] \tag{2.45b}$$

Iterative resolution Several iterative loops are used in order to solve this unsteady nonlinear coupled system. As shown in Figure 3.4, the first iterative loop to consider is the ***TIME loop*** (time-stepping). It uses the n-indexing. The loop starts with the knowledge of the numerical field solved at the previous (n-1)$^{\mathrm{th}}$ time-step and ends with the numerical solution at the current n$^{\mathrm{th}}$ time-step. During one temporal iteration, several iterative loops are needed for solving the coupled equations. The main one is the ***PIMPLE loop***. It uses the m-indexing. It contains the second one that is the ***PISO loop*** that uses the k-indexing. Each loop is divided into different steps. At any step of the resolution, all fields, excepted the calculated one, are considered as fixed terms inside the treated equations. To perform the numerical resolution, the discretization schemes introduced in Section 3.1 and Section 3.2 are used. All the steps identified in Fig. 3.4 are detailed in the following sub-sections.

Notations used Table 3.2 indicates the notations that will be used in this section. Given a field ψ, $\mathscr{I}\left(\psi_{\mathrm{P}}\right)_{\mathrm{f}}$ represent the discretization scheme used to interpolate the numerical value ψ_{P} at the center of a face f (see Sec. 3.1.3). ψ_{f} refers to the numerical value obtained at the center of f. Defining $\widetilde{\psi}_{\mathrm{f}} = \mathscr{I}\left(\psi_{\mathrm{P}}\right)_{\mathrm{f}}$, in most cases $\widetilde{\psi}_{\mathrm{f}} \neq \psi_{\mathrm{f}}$ because of some additional numerical corrections [3]. The schemes related to the operator $\mathscr{I}\left(\right)_{\mathrm{f}}$ are chosen by the user. The schemes are chosen according to the equation in which they are used and the fields to which they are applied. The same remark can be made for ψ_{P} and $\mathscr{I}\left(\psi_{\mathrm{f}}\right)_{\mathrm{P}}$. The schemes are those presented

3. More details are given in the following.

in Sec. 3.1.

In this document, the indexes used for the iterative loops obey to specific rules. Given an iterative loop using an a-indexing:

— the initial value of the loop gets the 0^{th} index;
— the first step of a loop gets the 1^{st} index;
— the last step of a loop gets the $a_{\text{last}}{}^{\text{th}}$ index;
— given sub-loop using the b-indexing and given a numerical field ψ, without additional indication, the updated value of ψ in the a-loop equals the last update of ψ in the b-loop $\psi^{(\ldots,a)} = \psi^{(\ldots,a,b_{\text{last}})}$;
— without additional indication, the initial value of ψ in sub-loop equals the value of ψ at the end of the previous a-loop $\psi^{(\ldots,a,0)} = \psi^{(\ldots,a\text{-}1)}$.

As mentioned, the main iterative loops are the the Time-loop with n-indexing, PIMPLE-loop with m-indexing and the PISO loop with k-indexing. For other loops described in this document, the q-indexing will be used. With q only referring to the considered loop in the corresponding section.

Table 3.2 – Synthesis of notations used for the spatial discretization of a field ψ at a cell P

Numerical values	**Approximated physical values**	**Schemes**	**Involved variables**
ψ_P	ψ at cell-center P		
ψ_f	ψ at a face-center		
$\nabla\psi\vert_P$	$\nabla\psi$ at cell-center P	Sec. 3.1.8	ψ_P, ψ_N (all neighbours of cell P)
$\nabla\psi\vert_f$	$\nabla\psi$ at a face-center	Sec. 3.1.6	ψ_P, ψ_N (neighbour sharing face f)
$\nabla\bullet\psi\vert_P$	$\nabla\bullet\psi$ at cell-center P	Sec. 3.1.5	ψ_P, ψ_N (all neighbours of cell P)
$\dfrac{\partial\psi_P}{\partial t}$	$\dfrac{\partial\psi}{\partial t}$ at ⊂cell-center P	Sec. 3.2	$\psi_P^{(n)}$, $\psi_P^{(n\text{-}1)}$, $\psi_P^{(n\text{-}2)}$
$\mathscr{I}\left(\psi_P\right)_f$	ψ at a face-center	Sec. 3.1.3	ψ_P, ψ_N (neighbour sharing face f)
$\mathscr{I}\left(\psi_f\right)_P$	ψ at cell-center P	Sec. 3.1.4	ψ_f (all faces of cell P)

In the following, the equations used for the update of main fields are identified by red or blue terms. The colored term is the one that is calculated by the equation. The blue color is used when the equation directly gives a definition of the numerical variable. The red color is used when the equation leads to a matrix system that is solved with a linear solver as detailed in Section 3.3.7.

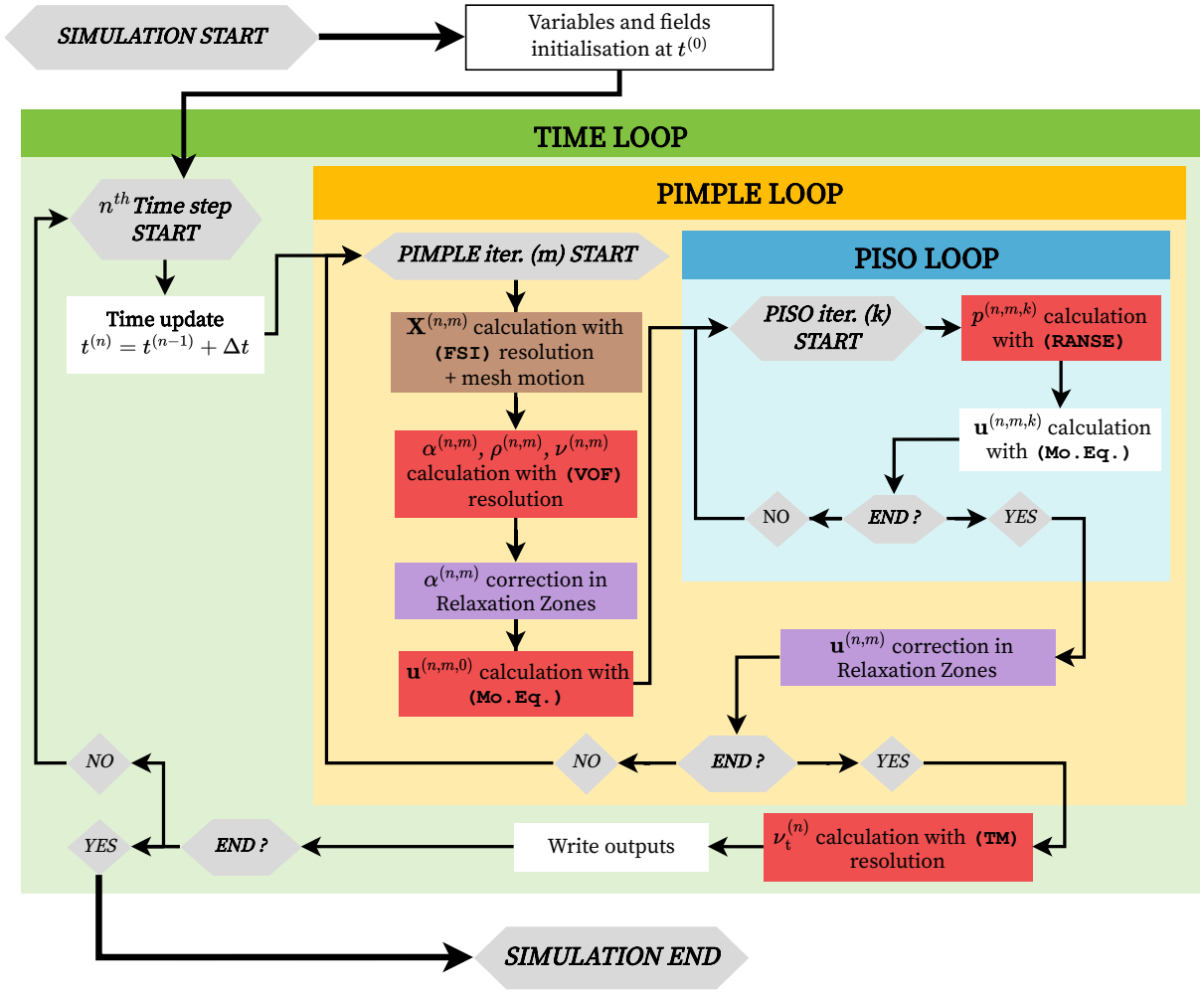


Figure 3.4 – *foamStar* simplified flowchart

3.3.1 Step one of PIMPLE loop: Body and mesh motion calculation

The first step of the PIMPLE loop is the body motion calculation and the resulting mesh motion. Table 3.3 indicates the main fields updated during this step.

Table 3.3 – Numerical fields updated by the FSI resolution

Numerical values	Description
$\mathbf{F}_{\text{body}}^{(\text{n,m})}$	External forces over the body
$\forall f \in \partial\Omega_{body}, \quad \mathbf{u}_{\text{f}}$	Velocity of the faces of the body
$\mathbf{X}_{\text{c}}^{(\text{n,m})}$, $\mathbf{X}_{\text{P}}^{(\text{n,m})}$, $\mathbf{X}_{\text{F}}^{(\text{n,m})}$	Positions of mesh points (cell corners, cell centers, face centers)
$V_{\text{P}}^{(\text{n,m})}$	Cells volume
$\mathbf{n}_{\text{f}}{}^{(\text{n,m})}$, $S_{\text{f}}^{(\text{n,m})}$, $\mathbf{S}_{\text{f}}^{(\text{n,m})}$	Normal vector, area and surface vector of faces
$\phi_{\text{mf}}^{(\text{n,m})}$	Faces flux due to mesh motion

a) FSI resolution with 6 DoFs rigid body motion

The equations needed for the body motion resolution are recalled here:

$$\overline{\overline{\Sigma}} = -\left[p + \rho \mathbf{g} \bullet \mathbf{x}\right] \overline{\overline{\mathrm{I}}} + \mu_{\text{eff}} \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right) \tag{2.44}$$

$$\mathbf{F}_{\text{fluid}} = \int_{\delta\Omega_{body}} \mathbf{dS} \bullet \overline{\overline{\Sigma}} \tag{2.43a}$$

$$\mathbf{M}_{\text{fluid}}^{\text{b}} = \mathbf{R}_0^{\text{b}} \left[\int_{M \in \delta\Omega_{body}} \mathbf{dS} \bullet \overline{\overline{\Sigma}} \wedge \mathbf{C_g M} \right] \tag{2.43b}$$

$$m\ddot{\mathbf{X}}_{\text{Cg}} = m\mathbf{g} + \mathbf{F}_{\text{fluid}} + \mathbf{F}_{\text{add}} + \mathbf{F}_{\text{c}} \tag{2.41}$$

$$\mathbf{I}_{\text{g}} \dot{\boldsymbol{\omega}}^{\text{b}} = \mathbf{M}_{\text{fluid}}^{\text{b}} + \mathbf{M}_{\text{add}}^{\text{b}} + \boldsymbol{\omega}^{\text{b}} \wedge \left(\mathbf{I}_{\text{g}} \boldsymbol{\omega}^{\text{b}} \right) \tag{2.42}$$

$$\forall M \in \partial\Omega_{body}, \quad \mathbf{X}_{\text{M}} = \mathbf{X}_{\text{Cg}} + \mathbf{R}_{\text{b}}^0 \left(\mathbf{C_g M}^{\text{b}} \right) \tag{2.45a}$$

$$\mathbf{u}_{\text{wall}}(M) = \dot{\mathbf{X}}_{\text{Cg}} + \mathbf{R}_{\text{b}}^0 \left[\mathbf{C_g M}^{\text{b}} \wedge \boldsymbol{\omega}^{\text{b}} \right] \tag{2.45b}$$

The second Newton law (Eq. 2.41) is valid in a Galilean reference frame. However, in *foamStar*, the motions, velocities and accelerations $\mathbf{X}$, $\dot{\mathbf{X}}$, $\ddot{\mathbf{X}}$ are written and derivated in the moving-domain reference frame $\mathscr{R}_{\text{D}}$. Consequently, this equation is only valid for constant domain-velocity $\mathbf{v}_0$ or for imposed acceleration in the $\mathbf{v}_0$ direction. The discretized forms of these equation are,

$$\forall f \in \partial\Omega_{body},\ \overline{\overline{\Sigma_{\mathrm{f}}}}^{(\mathrm{n,m})} = \left(-\left[p_{\mathrm{f}} + \rho_{\mathrm{f}}\mathbf{g}\bullet\mathbf{X}_{\mathrm{f}}\right]\bar{\bar{\mathrm{I}}} + \mu_{\mathrm{eff}\,\mathrm{f}}\left(\nabla\mathbf{u}\big|_{\mathrm{f}} + \nabla\mathbf{u}\big|_{\mathrm{f}}^{T}\right)\right)^{(\mathrm{n,m})} \tag{3.55}$$

$$\mathbf{F}_{\mathrm{fluid}}^{(\mathrm{n,m})} = \sum_{f\in\partial\Omega_{body}} \left(\mathbf{S}_{\mathrm{f}}\bullet\overline{\overline{\Sigma_{\mathrm{f}}}}\right)^{(\mathrm{n,m\text{-}1})} \tag{3.56}$$

$$\mathbf{M}_{\mathrm{fluid}}^{\mathrm{b}\ (\mathrm{n,m})} = \mathbf{R}_0^{\mathrm{b}(\mathrm{n,m})}\left[\sum_{f\in\partial\Omega_{body}} \mathbf{S}_{\mathrm{f}}\bullet\overline{\overline{\Sigma}} \wedge \mathbf{C}_{\mathbf{g}}\mathbf{M}\right]^{(\mathrm{n,m\text{-}1})} \tag{3.57}$$

$$\mathbf{F}_{\mathrm{tot}}^{(\mathrm{n,m})} = m\mathbf{g} + \mathbf{F}_{\mathrm{fluid}}^{(\mathrm{n,m})} + \mathbf{F}_{\mathrm{add}}^{(\mathrm{n,m})} \tag{3.58}$$

$$\mathbf{M}_{\mathrm{tot}}^{\mathrm{b}\ (\mathrm{n,m})} = \mathbf{M}_{\mathrm{fluid}}^{\mathrm{b}\ (\mathrm{n,m})} + \mathbf{M}_{\mathrm{add}}^{\mathrm{b}\ (\mathrm{n,m})} + \boldsymbol{\omega}^{\mathrm{b}(\mathrm{n,m})} \wedge \left(\boldsymbol{\omega}^{\mathrm{b}(\mathrm{n,m})}\mathbf{I}_{\mathrm{g}}\right) \tag{3.59}$$

$$m\ddot{\mathbf{X}}_{\mathrm{Cg}}^{(\mathrm{n,m})} = \mathbf{F}_{\mathrm{tot}}^{(\mathrm{n,m})} + \mathbf{F}_{\mathrm{c}}^{(\mathrm{n,m})} \tag{3.60}$$

$$\dot{\boldsymbol{\omega}}^{\mathrm{b}\ (\mathrm{n,m})}\mathbf{I}_{\mathrm{g}} = \mathbf{M}_{\mathrm{tot}}^{\mathrm{b}\ (\mathrm{n,m})} + \mathbf{M}_{\mathrm{c}}^{\mathrm{b}(\mathrm{n,m})} \tag{3.61}$$

Constraints To take into account constraints and evaluate $\mathbf{F_c}^{(\mathrm{n,m})}$, a method based on Lagrange multipliers is used. It consists in constraining the direction of accelerations $\ddot{\mathbf{X}}_{\mathrm{Cg}}$ and $\dot{\boldsymbol{\omega}}^{\mathrm{b}}$. From the 0$^{\mathrm{th}}$ time-step to the N$^{\mathrm{th}}$ time-step, a system of constraints is defined as two vector functions defining the directions of the constraints,

$$\begin{array}{rcl} \mathbf{C}_{\mathrm{F}}: & [|0,N|] \longrightarrow \mathbb{R}^3 & \quad \text{and} \quad \mathbf{C}_{\mathrm{M}}: \quad [|0,N|] \longrightarrow \mathbb{R}^3 \\ & n \longmapsto \mathbf{C}_{\mathrm{F}}^{(\mathrm{n})} & \qquad\qquad\quad n \longmapsto \mathbf{C}_{\mathrm{M}}^{(\mathrm{n})} \end{array} \tag{3.62}$$

and two scalar functions defining the magnitudes of the constraints,

$$\begin{array}{rcl} \gamma_{\mathrm{F}}: & [|0,N|] \longrightarrow \mathbb{R} & \quad \text{and} \quad \gamma_{\mathrm{M}}: \quad [|0,N|] \longrightarrow \mathbb{R} \\ & n \longmapsto \gamma_{\mathrm{F}}^{(\mathrm{n})} & \qquad\qquad\quad n \longmapsto \gamma_{\mathrm{M}}^{(\mathrm{n})} \end{array} \tag{3.63}$$

With these imposed quantities, the constraints are represented by the following relations,

$$\forall n \in [|0,N|], \quad \frac{\mathbf{C}_{\mathrm{F}}^{(\mathrm{n})}}{\left|\left|\mathbf{C}_{\mathrm{F}}^{(\mathrm{n})}\right|\right|}\bullet\ddot{\mathbf{X}}_{\mathrm{Cg}}^{(\mathrm{n,m})} = \gamma_{\mathrm{F}}^{(\mathrm{n})} \quad \text{and} \quad \frac{\mathbf{C}_{\mathrm{M}}^{(\mathrm{n})}}{\left|\left|\mathbf{C}_{\mathrm{M}}^{(\mathrm{n})}\right|\right|}\bullet\dot{\boldsymbol{\omega}}^{\mathrm{b}\ (\mathrm{n,m})} = \gamma_{\mathrm{M}}^{(\mathrm{n})} \tag{3.64}$$

To impose the constraints in Eq. 2.41 and Eq. 2.42, the constraints force $\mathbf{F}_{\mathrm{c}}^{(\mathrm{n,m})}$ and moments $\mathbf{M}_{\mathrm{c}}^{\mathrm{b}(\mathrm{n,m})}$ are defined so that they force the components of the total forces $\mathbf{F}_{\mathrm{tot}}^{(\mathrm{n,m})}$ and moments $\mathbf{M}_{\mathrm{tot}}^{\mathrm{b}\ (\mathrm{n,m})}$ to be along the constraint directions. This results in the following formulation,

$$\mathbf{F}_{\mathrm{c}}^{(\mathrm{n,m})} = \left[m\gamma_{\mathrm{F}}^{(\mathrm{n})} - \mathbf{F}_{\mathrm{tot}}^{(\mathrm{n,m})} \cdot \frac{\mathbf{C}_{\mathrm{M}}^{(\mathrm{n})}}{\left|\left|\mathbf{C}_{\mathrm{M}}^{(\mathrm{n})}\right|\right|} \right] \frac{\mathbf{C}_{\mathrm{M}}^{(\mathrm{n})}}{\left|\left|\mathbf{C}_{\mathrm{M}}^{(\mathrm{n})}\right|\right|} \tag{3.65a}$$

$$\mathbf{M}_{\mathrm{c}}^{\mathrm{b}(\mathrm{n,m})} = \mathbf{I}_{\mathrm{g}} \left[\gamma_{\mathrm{M}}^{(\mathrm{n})} - \left(\mathbf{I}_{\mathrm{g}}^{-1} \mathbf{M}_{\mathrm{tot}}^{\mathrm{b}\ (\mathrm{n,m})} \right) \cdot \frac{\mathbf{C}_{\mathrm{M}}^{(\mathrm{n})}}{\left|\left|\mathbf{C}_{\mathrm{M}}^{(\mathrm{n})}\right|\right|} \right] \frac{\mathbf{C}_{\mathrm{M}}^{(\mathrm{n})}}{\left|\left|\mathbf{C}_{\mathrm{M}}^{(\mathrm{n})}\right|\right|} \tag{3.65b}$$

Resolution The resolution of the system of equations is done defining a vector $\mathbf{Y}$ and its time derivative $\dot{\mathbf{Y}}$ as,

$$\mathbf{Y}^{(\mathrm{n,m})} = \begin{bmatrix} \mathbf{X}_{\mathrm{Cg}} \\ \dot{\mathbf{X}}_{\mathrm{Cg}} \\ \omega^{\mathrm{b}} \\ \Theta \end{bmatrix}^{(\mathrm{n,m})}, \quad \dot{\mathbf{Y}}^{(\mathrm{n,m})} = \begin{bmatrix} \dot{\mathbf{X}}_{\mathrm{Cg}} \\ \ddot{\mathbf{X}}_{\mathrm{Cg}} \\ \dot{\omega}^{\mathrm{b}} \\ \dot{\Theta} \end{bmatrix}^{(\mathrm{n,m})} \tag{3.66}$$

Then, the equations are set into matrix $\mathbf{M}^{(\mathrm{n,m})}$ and vector $\mathbf{S}^{(\mathrm{n,m})}$ defined by,

$$\mathbf{M}^{(\mathrm{n,m})} = \begin{bmatrix} \bar{\bar{\mathbf{0}}} & \bar{\bar{\mathbf{I}}} & \bar{\bar{\mathbf{0}}} & \bar{\bar{\mathbf{0}}} \\ \bar{\bar{\mathbf{0}}} & \bar{\bar{\mathbf{0}}} & \bar{\bar{\mathbf{0}}} & \bar{\bar{\mathbf{0}}} \\ \bar{\bar{\mathbf{0}}} & \bar{\bar{\mathbf{0}}} & \bar{\bar{\mathbf{0}}} & \bar{\bar{\mathbf{0}}} \\ \bar{\bar{\mathbf{0}}} & \bar{\bar{\mathbf{0}}} & \mathbf{T}^{(\mathrm{n,m})} & \bar{\bar{\mathbf{0}}} \end{bmatrix}, \quad \mathbf{S}^{(\mathrm{n,m})} = \begin{bmatrix} \mathbf{0} \\ \frac{1}{m}\left[\mathbf{F}_{\mathrm{tot}} + \mathbf{F}_{\mathrm{c}}\right]^{(\mathrm{n,m})} \\ \mathbf{I}_{\mathrm{g}}^{-1}\left[\mathbf{M}_{\mathrm{tot}}^{\mathrm{b}} + \mathbf{M}_{\mathrm{c}}^{\mathrm{b}}\right]^{(\mathrm{n,m})} \\ \mathbf{0} \end{bmatrix} \tag{3.67}$$

where,

$$\mathbf{T}^{(\mathrm{n,m})} = \begin{bmatrix} 1 & \sin\phi^{(\mathrm{n,m})} \tan\theta^{(\mathrm{n,m})} & \cos\phi^{(\mathrm{n,m})} \tan\theta^{(\mathrm{n,m})} \\ 0 & \cos\phi^{(\mathrm{n,m})} & -\sin\phi^{(\mathrm{n,m})} \\ 0 & \sin\phi^{(\mathrm{n,m})} / \cos\theta^{(\mathrm{n,m})} & \cos\phi^{(\mathrm{n,m})} / \cos\theta^{(\mathrm{n,m})} \end{bmatrix} \tag{3.68}$$

With this formulation, the system of equation becomes,

$$\dot{\mathbf{Y}}^{(\mathrm{n,m})} = \mathbf{M}^{(\mathrm{n,m})} \mathbf{Y}^{(\mathrm{n,m})} + \mathbf{S}^{(\mathrm{n,m})} \tag{3.69}$$

Finally after the resolution of this system [4], the updated positions of all the mesh corners c and face centers F on the body-wall are,

$$\forall c \in \partial\Omega_{body}, \quad \mathbf{X}_{\mathrm{c}}^{(\mathrm{n,m})} = \mathbf{X}_{\mathrm{Cg}}^{(\mathrm{n,m})} + \mathbf{R}_{\mathrm{b}}^{0(\mathrm{n,m})} \left(\boldsymbol{\xi}_{\mathrm{c}}^{\mathrm{b}}\right) \tag{3.70a}$$

$$\forall f \in \partial\Omega_{body}, \quad \mathbf{X}_{\mathrm{F}}^{(\mathrm{n,m})} = \mathbf{X}_{\mathrm{Cg}}^{(\mathrm{n,m})} + \mathbf{R}_{\mathrm{b}}^{0(\mathrm{n,m})} \left(\boldsymbol{\xi}_{\mathrm{F}}^{\mathrm{b}}\right) \tag{3.70b}$$

$$\tag{3.70c}$$

and the velocities at the face center are imposed by:

$$\forall f \in \partial\Omega_{body}, \quad \mathbf{u}_{\mathrm{f}}^{(\mathrm{n,m})} = \dot{\mathbf{X}}_{\mathrm{Cg}}^{(\mathrm{n,m})} + \mathbf{R}_{\mathrm{b}}^{0(\mathrm{n,m})} \left[\boldsymbol{\xi}_{\mathrm{F}}^{\mathrm{b}} \wedge \boldsymbol{\omega}^{\mathrm{b}(\mathrm{n,m})}\right] \tag{3.71}$$

b) Mesh morphing

At this step, the positions $\mathbf{X}_{\mathrm{c}}$ of mesh-points and $\mathbf{X}_{\mathrm{F}}$ of faces are known on the body-walls $\partial\Omega_{body}$. Using these boundary conditions, the entire mesh is morphed in order to smooth the deformation from the body to the exterior boundaries. The exterior boundaries are kept fixed in reference frame $\mathscr{R}_0$. For all faces f on boundaries, the displacement of their face centers and face corners c are defined as,

$$\boldsymbol{\delta}_{\mathrm{F}}^{(\mathrm{n,m})} = \mathbf{X}_{\mathrm{F}}^{(\mathrm{n,m})} - \mathbf{X}_{\mathrm{F}}^{(\mathrm{n,m\text{-}1})} \tag{3.72a}$$

$$\boldsymbol{\delta}_{\mathrm{c}}^{(\mathrm{n,m})} = \mathbf{X}_{\mathrm{c}}^{(\mathrm{n,m})} - \mathbf{X}_{\mathrm{c}}^{(\mathrm{n,m\text{-}1})} \tag{3.72b}$$

Then, the cell centers displacements $\boldsymbol{\delta}_{\mathrm{P}}^{(\mathrm{n,m})}$ are computed solving the following Poisson equation:

$$\nabla\bullet\left(\Gamma(\mathbf{X}_{\mathrm{P}}^{(\mathrm{n,m\text{-}1})})\,\nabla\boldsymbol{\delta}|_{\mathrm{P}}^{(\mathrm{n,m})}\right) = 0 \tag{3.73}$$

with $\Gamma(\mathbf{X}_{\mathrm{P}}^{(\mathrm{n,m\text{-}1})})$ the diffusion coefficient that can take several forms depending on the desired aspect of the mesh deformation propagation. In *foamStar*, this coefficient is the square of the inverse distance $r_{\mathrm{P}}^{(\mathrm{n,m\text{-}1})}$ from the body-wall to the point P by default:

$$\Gamma(\mathbf{X}_{\mathrm{P}}^{(\mathrm{n,m\text{-}1})}) = \left(\frac{1}{r_{\mathrm{P}}^{(\mathrm{n,m\text{-}1})}}\right)^2 \tag{3.74}$$

After the resolution of this Poisson equation, the cell corners displacements are obtained

4. In *foamStar*, a Runge-Kutta-like algorithm is commonly used.

interpolating the cell centers displacements. Finally, the positions of all the inner cell corners are obtained applying Eq. 3.72b. After this update, the cells corner are kept fixed for the rest of the PIMPLE iteration.

Knowing the new positions of the cell corners, the positions $\mathbf{X}_\mathrm{F}$, $\mathbf{X}_\mathrm{P}$ and the variables $S_\mathrm{f}^{(\mathrm{n,m})}$, $\mathbf{n}_\mathrm{f}{}^{(\mathrm{n,m})}$, $\mathbf{S}_\mathrm{f}^{(\mathrm{n,m})}$, $V_\mathrm{P}^{(\mathrm{n,m})}$, $\phi_\mathrm{mf}^{(\mathrm{n,m})}$ can be updated. Firstly, as shown in the Figure 3.5, to get the position of the geometrical face center F and the surface vector $\mathbf{S}_\mathrm{f}$, the considered face f is divided into triangles formed by each edge of the face and the barycenter $\widetilde{\mathrm{F}}$ of the N_V vertexes $\mathrm{V_i}$ defined as:

$$\mathbf{X}_{\widetilde{\mathrm{F}}} = \frac{1}{N_\mathrm{V}} \sum_{i=1}^{N_\mathrm{V}} \mathbf{X}_{\mathrm{V_i}} \tag{3.75}$$

Then, the positions of geometrical centers $\mathrm{T_i}$ and surface vectors $\mathbf{S}_{\mathrm{f}i}$ of each triangle are obtained as follows:

$$\begin{aligned}
\text{with} \quad & \mathrm{V}_{\mathrm{N_V+1}} \equiv \mathrm{V}_1 \\
& \forall i \in [|1, N_\mathrm{V}|],\ \mathbf{X}_{\mathrm{T_i}} = \frac{1}{3}\left[\mathbf{X}_{\widetilde{\mathrm{F}}} + \mathbf{X}_{\mathrm{V_i}} + \mathbf{X}_{\mathrm{V_{i+1}}}\right] && \text{(3.76a)} \\
& \forall i \in [|1, N_\mathrm{V}|],\ \mathbf{S}_{\mathrm{f}i} = \frac{1}{2}\widetilde{\mathbf{F}}\mathbf{V_i} \wedge \widetilde{\mathbf{F}}\mathbf{V_{i+1}} && \text{(3.76b)}
\end{aligned}$$

Finally,

$$\mathbf{S}_\mathrm{f} = \sum_{i=1}^{N_\mathrm{V}} \mathbf{S}_{\mathrm{f}i} \tag{3.77a}$$

$$S_\mathrm{f} = |\mathbf{S}_\mathrm{f}| \tag{3.77b}$$

$$\mathbf{n}_\mathrm{f} = \frac{\mathbf{S}_\mathrm{f}}{S_\mathrm{f}} \tag{3.77c}$$

$$\mathbf{X}_\mathrm{F} = \frac{1}{S_\mathrm{f}} \sum_{i=1}^{N_\mathrm{V}} |\mathbf{S}_{\mathrm{f}i}|\, \mathbf{X}_{\mathrm{T_i}} \tag{3.77d}$$

$$\tag{3.77e}$$

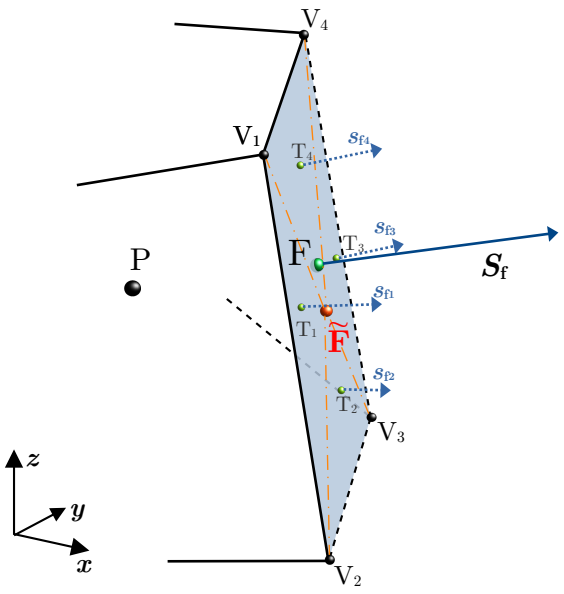


Figure 3.5 – Face decomposition used for the calculations of surface areas and geometric-center of faces. With F the geometric center and $\widetilde{\mathrm{F}}$ the barycenter of vertices.

To compute the position of the cell center P and the cell volume V_P, a similar approach is done decomposing the cell into pyramids for which, the base is a face $\mathrm{f_i}$ and the apex is the barycentrer $\widetilde{\mathrm{P}}$ of the N_f faces center $\mathrm{F_i}$. This decomposition is shown in Figure 3.6. The resulting equation is:

$$\mathbf{X}_{\widetilde{\mathrm{P}}} = \frac{1}{N_\mathrm{f}} \sum_{i=1}^{N_\mathrm{f}} \mathbf{X}_{\mathrm{F_i}} \tag{3.78}$$

Then, the geometrical center $\mathrm{P_i}$ and volume V_i of each pyramid are obtained with,

$$\forall i \in [|1, N_\mathrm{f}|],\ \mathbf{X}_{\mathrm{P_i}} = \mathbf{X}_{\widetilde{\mathrm{P}}} + \frac{3}{4}\widetilde{\mathbf{PF_i}} \tag{3.79a}$$

$$\forall i \in [|1, N_\mathrm{f}|],\ V_i = \frac{1}{3}\widetilde{\mathbf{PF_i}} \bullet \mathbf{S}_{\mathrm{f}i} \tag{3.79b}$$

Finally,

$$V_\mathrm{P} = \sum_{i=1}^{N_\mathrm{f}} V_i \tag{3.80a}$$

$$\mathbf{X}_\mathrm{P} = \frac{1}{V_\mathrm{P}} \sum_{i=1}^{N_\mathrm{f}} V_i \mathbf{X}_{\mathrm{P_i}} \tag{3.80b}$$

$$\tag{3.80c}$$

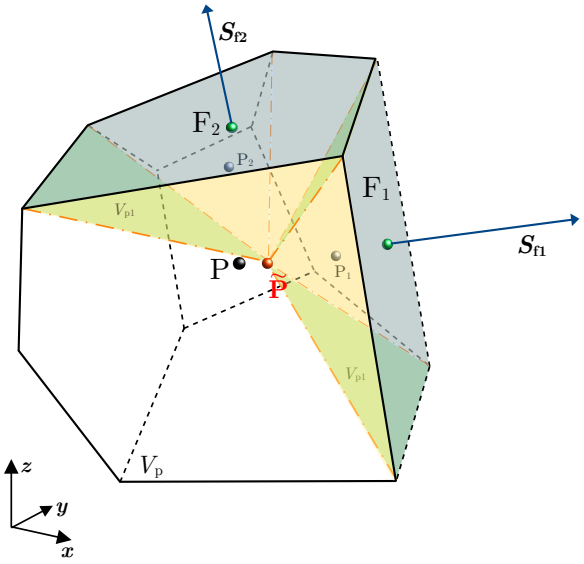


Figure 3.6 – Cell decomposition used for the calculations of volume and geometric-center of cells. With P the geometric center and $\widetilde{P}$ the barycenter of face centers.

As seen previously with Eq. 3.54, the mesh flux $\phi_{mf}^{(n,m)}$ of a face f depends on the variation of the cell volume. Consequently, to evaluate $\phi_{mf}^{(n,m)}$, the "swept volume" Vs_f of the face has to be estimated. To do so, the face is decomposed into triangles formed by each edge and the geometrical center F of the face. This decomposition is done for the face at the previous time-step (n-1) and at the current time-step and PIMPLE-step (n,m) as shown in Figure 3.7.

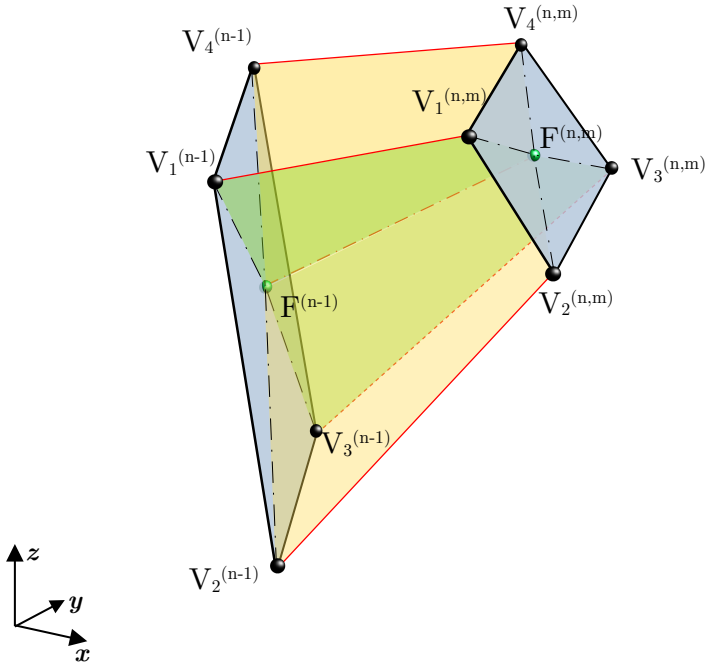


Figure 3.7 – Face-swept volume decomposition used for mesh flux computation

In the 4D space (t,x,y,z) the volume formed by a sweeping triangle between steps (n-1) and (n,m) is not know so it has to be approximated. Figure 3.8 shows a representation of this volume. The triangles considered here are FV_1V_2 at steps (n-1) and (n,m). The swept volume Vs_T of the considered triangle is evaluated considering 3 tetrahedrons. Such a structure is not unique and because the lateral quadrilateral faces are not necessarily planar, the sum of the volumes of the tetrahedrons would depend on the chosen structure. To manage this problem, the swept volume Vs_T is the mean of the volumes of two structures of tetrahedrons. For the first one shown in Figure 3.8(a) the face $V_1^{(n-1)}V_2^{(n-1)}V_2^{(n,m)}V_1^{(n,m)}$ is decomposed considering the edge $V_1^{(n-1)}V_2^{(n,m)}$. For the second one it is the edge $V_2^{(n-1)}V_1^{(n,m)}$ that is involved. Such an alternative tetrahedrons organization is not needed for the other lateral faces because they are

internal faces from decomposition of the face-swept volume shown in Figure 3.8(b).

$$\mathrm{V}_{\mathrm{N_V}+1} \equiv \mathrm{V}_1 \tag{3.81a}$$

$$\begin{aligned} \forall i \in [|1, N_\mathrm{V}|], \quad Vs_{\mathrm{T_i}} &= \frac{1}{6}\mathbf{F}^{(\mathrm{n\text{-}1})}\mathbf{F}^{(\mathrm{n,m})} \cdot \left(\mathbf{F}^{(\mathrm{n\text{-}1})}\mathbf{V}_\mathrm{i}^{(\mathrm{n\text{-}1})} \wedge \mathbf{F}^{(\mathrm{n\text{-}1})}\mathbf{V}_{\mathrm{i+1}}^{(\mathrm{n\text{-}1})}\right) \\ &+ \frac{1}{12}\mathbf{V}_{\mathrm{i+1}}^{(\mathrm{n\text{-}1})}\mathbf{V}_{\mathrm{i+1}}^{(\mathrm{n,m})} \cdot \left(\mathbf{V}_{\mathrm{i+1}}\mathbf{F}^{(\mathrm{n,m})} \wedge \mathbf{V}_{\mathrm{i+1}}^{(\mathrm{n,m})}\mathbf{V}_\mathrm{i}^{(\mathrm{n,m})}\right) \\ &+ \frac{1}{12}\mathbf{V}_\mathrm{i}^{(\mathrm{n\text{-}1})}\mathbf{V}_{\mathrm{i+1}}^{(\mathrm{n\text{-}1})} \cdot \left(\mathbf{V}_\mathrm{i}^{(\mathrm{n\text{-}1})}\mathbf{F}^{(\mathrm{n,m})} \wedge \mathbf{V}_\mathrm{i}^{(\mathrm{n\text{-}1})}\mathbf{V}_\mathrm{i}^{(\mathrm{n,m})}\right) \\ &+ \frac{1}{12}\mathbf{V}_\mathrm{i}^{(\mathrm{n\text{-}1})}\mathbf{V}_\mathrm{i}^{(\mathrm{n,m})} \cdot \left(\mathbf{V}_\mathrm{i}^{(\mathrm{n,m})}\mathbf{V}_{\mathrm{i+1}}^{(\mathrm{n,m})} \wedge \mathbf{V}_1\mathbf{F}^{(\mathrm{n,m})}\right) \\ &+ \frac{1}{12}\mathbf{V}_{\mathrm{i+1}}^{(\mathrm{n\text{-}1})}\mathbf{V}_\mathrm{i}^{(\mathrm{n\text{-}1})} \cdot \left(\mathbf{V}_{\mathrm{i+1}}^{(\mathrm{n\text{-}1})}\mathbf{V}_{\mathrm{i+1}}^{(\mathrm{n,m})} \wedge \mathbf{V}_{\mathrm{i+1}}^{(\mathrm{n\text{-}1})}\mathbf{F}^{(\mathrm{n,m})}\right) \end{aligned} \tag{3.81b}$$

$$Vs_\mathrm{f} = \sum_{i=1}^{N_\mathrm{V}} Vs_{\mathrm{T_i}} \tag{3.81c}$$

With this formulation of Vs_f and the cell volume V_P, the following relation is verified:

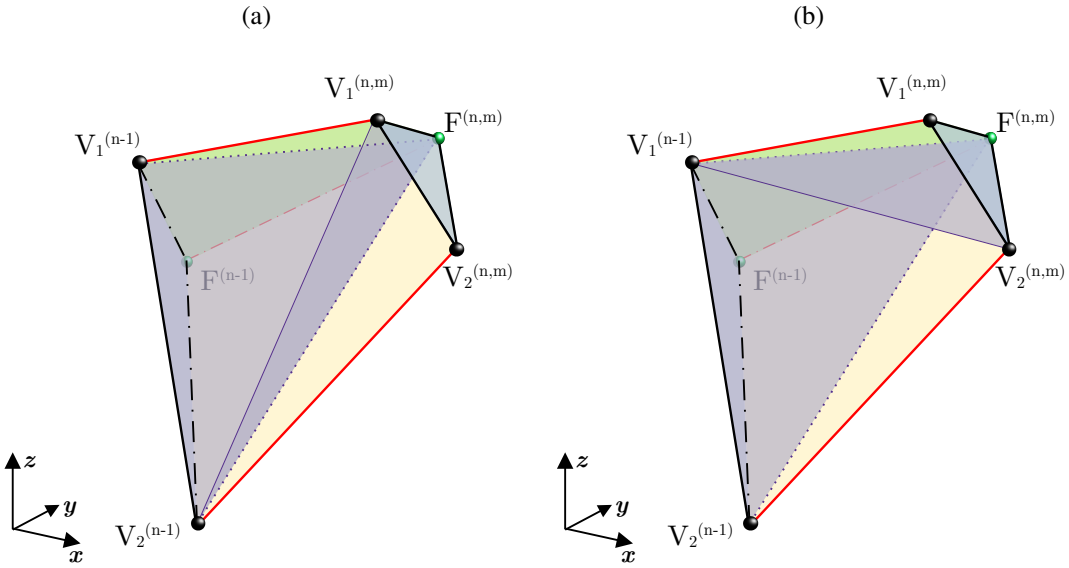
$$V_\mathrm{P}^{(\mathrm{n,m})} - V_\mathrm{P}^{(\mathrm{n\text{-}1})} = \sum_\mathrm{f} Vs_\mathrm{f}^{(\mathrm{n,m})} \tag{3.82}$$

Consequently, the discretized geometric conservation law (Eq. 3.54) implies the following expression of the mesh flux $\phi_\mathrm{mf}^{(\mathrm{n,m})}$ for the face f:

$$\phi_\mathrm{mf}^{(\mathrm{n,m})} = c_\mathrm{n}^{(\mathrm{n})} Vs_\mathrm{f}^{(\mathrm{n,m})} - c_{\mathrm{n\text{-}2}}^{(\mathrm{n})} Vs_\mathrm{f}^{(\mathrm{n\text{-}2})} + \gamma_{Vs_\mathrm{f}}^{(\mathrm{n,m})} \tag{3.83}$$

As mentioned in Section 3.2.3 the coefficients expression depends on the choice of the temporal discretization scheme.

Figure 3.8 – Two sub-face-swept decomposition required for mesh flux computation



3.3.2 Step two of PIMPLE loop: VOF resolution

Once the mesh morphing is updated, equations of the VOF model are solved. The set of equations in the moving-domain reference frame $\mathscr{R}_\mathrm{D}$ is,

$$\frac{\partial \alpha}{\partial t} + \nabla\bullet(\alpha\mathbf{u}) + \nabla\bullet\mathbf{u}_\mathrm{comp} = 0 \tag{2.18}$$

$$\rho = \alpha\rho_\mathrm{w} + (1-\alpha)\rho_\mathrm{a} \tag{2.5a}$$

$$\mu = \alpha\mu_\mathrm{w} + (1-\alpha)\mu_\mathrm{a} \tag{2.5b}$$

$$\nu = \frac{\mu}{\rho} \tag{2.5c}$$

Table 3.4 indicates the main fields updated during this step.

Table 3.4 – Numerical fields updated by the VOF resolution

Numerical values	Description
$\alpha_\mathrm{P}^{(n,m)}$	Volume fraction at cell centers
$\rho_\mathrm{P}^{(n,m)}$, $\mu_\mathrm{P}^{(n,m)}$, $\nu_\mathrm{P}^{(n,m)}$	Density, dynamic and kinematic viscosity of the VOF at cell centers
$F_{\rho_\mathrm{f}}^{(n,m)}$	Density flux at faces

Firstly, with the Finite Volume formulation, the discretized form of Eq. 2.18 at the n^{th}

time-step is:

$$c_{\mathrm{n}}^{(\mathrm{n})}\alpha_{\mathrm{P}}^{(\mathrm{n,m})}+S_{\mathrm{ddt}\alpha}^{(\mathrm{n})}+\sum_{\mathrm{f}}\left(\mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m})}\right)_{\mathrm{f}}\left[\phi_{\mathrm{f}}^{(\mathrm{n,m\text{-}1})}+\phi_{\mathrm{mf}}^{(\mathrm{n,m\text{-}1})}\right]+\phi_{\mathrm{comp.f}}^{(\mathrm{n,m})}\right)=0 \tag{3.84}$$

where $\phi_{\mathrm{comp.f}}^{(\mathrm{n,m})}$ is the additional compression flux at the face f. Its formulation is discussed in the following.

As defined in Section 3.2, the source term computed for the time derivative $S_{\mathrm{ddt}}^{(\mathrm{n})}$ corresponds to:

$$S_{\mathrm{ddt}\alpha}^{(\mathrm{n})}=c_{\mathrm{n\text{-}1}}^{(\mathrm{n})}V_{\mathrm{P}}^{(\mathrm{n\text{-}1})}\alpha_{\mathrm{P}}^{(\mathrm{n\text{-}1})}+c_{\mathrm{n\text{-}2}}^{(\mathrm{n})}V_{\mathrm{P}}^{(\mathrm{n\text{-}2})}\alpha^{(\mathrm{n\text{-}1})}+\gamma_{\alpha}^{(\mathrm{n})}$$

In *foamStar*, the VOF is solved by a recursive algorithm depending on the selected time-scheme. At the end of the resolution, the numerical α field respects the following equation:

$$c_{\mathrm{n}}^{(\mathrm{n})}V_{\mathrm{P}}^{(\mathrm{n})}\alpha_{\mathrm{P}}^{(\mathrm{n,m})}+S_{\mathrm{ddt}\alpha}^{(\mathrm{n})}+\sum_{\mathrm{f}}F_{\alpha_{\mathrm{f}}}^{(\mathrm{n,m})}=0 \tag{3.85}$$

The volume fraction flux $F_{\alpha_{\mathrm{f}}}$ is computed by the algorithm detailed in the following.

Considering the density field ρ defined by Eq. 2.5a a density-flux can be defined as:

$$\boxed{F_{\rho_{\mathrm{f}}}^{(\mathrm{n,m})}=(\rho_{\mathrm{w}}-\rho_{\mathrm{a}})F_{\alpha_{\mathrm{f}}}^{(\mathrm{n,m})}+\rho_{\mathrm{a}}\left[\phi_{\mathrm{f}}^{(\mathrm{n,m\text{-}1})}+\phi_{\mathrm{mf}}^{(\mathrm{n,m\text{-}1})}\right]} \tag{3.86}$$

This density flux is consistent with a simplified volume fraction flux:

$$\widetilde{F_{\alpha_{\mathrm{f}}}}^{(\mathrm{n,m})}=\alpha_{\mathrm{f}}^{(\mathrm{n,m})}\left[\phi_{\mathrm{f}}^{(\mathrm{n,m\text{-}1})}+\phi_{\mathrm{mf}}^{(\mathrm{n,m\text{-}1})}\right]$$

resulting in:

$$\widetilde{F_{\rho_{\mathrm{f}}}}^{(\mathrm{n,m})}=\rho_{\mathrm{f}}^{(\mathrm{n,m})}\left[\phi_{\mathrm{f}}^{(\mathrm{n,m\text{-}1})}+\phi_{\mathrm{mf}}^{(\mathrm{n,m\text{-}1})}\right]$$

Even with $F_{\alpha_{\mathrm{f}}}^{(\mathrm{n,m})}\neq\widetilde{F_{\alpha_{\mathrm{f}}}}^{(\mathrm{n,m})}$ (among other things because of the use of a compression term [5]) Eq. 3.85, Eq. 3.86 and Eq. 3.54 imply:

$$c_{\mathrm{n}}^{(\mathrm{n})}V_{\mathrm{P}}^{(\mathrm{n})}\rho_{\mathrm{P}}^{(\mathrm{n,m})}+S_{\mathrm{ddt}\rho}^{(\mathrm{n})}+\sum_{\mathrm{f}}F_{\rho_{\mathrm{f}}}^{(\mathrm{n,m})}=0 \tag{3.87}$$

that is then used as the discretized form of the mass conservation equation with compression term (Eq. 2.19).

In the following, the general resolution method is presented. Besides, the specific

5. More details can be found in the following.

implementations are detailed. Some discussions on the algorithm are provided in the next section (Sec. 3.3.3).

a) General formulation of the VOF resolution with *foamStar*

The numerical algorithm consists in solving the equations for the volume fraction with a q-indexed iterative procedure. It is initialized with the following predictor step:

$$c_{\mathrm{n}}^{(\mathrm{n})}\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}+\frac{1}{V_{\mathrm{P}}^{(\mathrm{n,m})}}\sum_{\mathrm{f}}\mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}\right)_{\mathrm{f}(UD)}\left[\phi_{\mathrm{f}}^{(\mathrm{n,m\text{-}1})}-\phi_{\mathrm{mf}}^{(\mathrm{n,m\text{-}1})}\right]=-\frac{S_{\mathrm{ddt}\alpha}^{(\mathrm{n})}}{V_{\mathrm{P}}^{(\mathrm{n,m})}} \tag{3.88a}$$

$$F_{\mathrm{L}}^{(0)}=\mathscr{I}_{\mathrm{UD}}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}\right)_{\mathrm{f}}\left[\phi_{\mathrm{f}}^{(\mathrm{n,m\text{-}1})}-\phi_{\mathrm{mf}}^{(\mathrm{n,m\text{-}1})}\right] \tag{3.88b}$$

At the initialisation step (q=0), α is predicted with the resolution of Eq. 3.88a. In this equation, the operator $\mathscr{I}_{\mathrm{UD}}(.)_{\mathrm{f}}$ represents an upwind spatial scheme (see Section 3.1.3). An initial low order flux $F_{\mathrm{L}}^{(0)}$ is defined by Eq. 3.88b. No VOF compression terms are used during this initialization step.

For $q>0$, the recursive Algorithm 3.89 consists in updating the flux $F_{\mathrm{L}}^{(\mathrm{q})}$ as a weighted sum of an high-order compressed volume fraction flux and the flux $F_{\mathrm{L}}^{(\mathrm{q\text{-}1})}$ calculated at the previous step. This step involves the weight factor $\lambda^{(\mathrm{q})}$ computed using Multidimensional Universal Limiter for Explicit Solution (MULES) algorithm detailed by Márquez Damián (2013). This method ensures the TVD criterion introduced in Section 3.1.3.

During the VOF resolution, $\alpha^{(\mathrm{n,m,q})}$ is computed with the following recursive formulation:

$$\phi_{\mathrm{comp.f}}^{(\mathrm{n,m,q})}=\mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,q\text{-}1})}\right)_{\mathrm{f}}\left[1-\mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,q\text{-}1})}\right)_{\mathrm{f}}\right]\phi_{\mathrm{r}}^{(\mathrm{n,m})} \tag{3.89a}$$

$$F_{\mathrm{H}}^{(\mathrm{q})}=\mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,q\text{-}1})}\right)_{\mathrm{f}}\left[\phi_{\mathrm{f}}^{(\mathrm{n,m\text{-}1})}-\phi_{\mathrm{mf}}^{(\mathrm{n,m\text{-}1})}\right]+\phi_{\mathrm{comp.f}}^{(\mathrm{n,m,q})} \tag{3.89b}$$

$$F_{\mathrm{L}}^{(\mathrm{q})}=(1-\lambda^{(\mathrm{q})})F_{\mathrm{L}}^{(\mathrm{q\text{-}1})}+\lambda^{(\mathrm{q})}F_{\mathrm{H}}^{(\mathrm{q})} \tag{3.89c}$$

$$\alpha_{\mathrm{P}}^{(\mathrm{n,m,q})}=\alpha_{\mathrm{P}}^{(\mathrm{n,m,q\text{-}1})}-\frac{1}{c_{\mathrm{n}}^{(\mathrm{n})}V_{\mathrm{P}}^{(\mathrm{n,m})}}\sum_{\mathrm{f}}\lambda^{(\mathrm{q})}(F_{\mathrm{H}}^{(\mathrm{q})}-F_{\mathrm{L}}^{(\mathrm{q\text{-}1})}) \tag{3.89d}$$

with $\phi_{\mathrm{r}}^{(\mathrm{n,m})}$ defined as,

$$\phi_{\mathrm{r}}^{(\mathrm{n,m})}=\gamma_{\mathrm{comp}}\left|\phi_{\mathrm{f}}^{(\mathrm{n,m\text{-}1})}-\phi_{\mathrm{mf}}^{(\mathrm{n,m\text{-}1})}\right|\frac{\nabla\alpha|_{\mathrm{f}}^{(\mathrm{n,m,q\text{-}1})}}{\left|\left|\nabla\alpha|_{\mathrm{f}}^{(\mathrm{n,m,q\text{-}1})}\right|\right|}\cdot\mathbf{n}_{\mathrm{f}}^{(\mathrm{n,m})}, \tag{3.90}$$

and γ_{comp} a positive constant.

As mentioned in Section 2.1.2, the compression term in Eq. 3.89b will only get non null

value in the free-surface zone with $0 < \alpha^{(n,m,q)} < 1$ and still verifies the mass conservation. However, because of the its nonlinear aspect, it has to be evaluate explicitly. This explicit formulation implies some instabilities for high values of γ_{comp}. The literature suggests to use $\gamma_{comp} \leq 1$ to preserve the numerical stability (OpenFOAM, 2022a; Choi et al., 2018). With $\gamma_{comp} = 0$, no additional compression term is used.

With this general formulation of the VOF resolution [6] the volume fraction flux F_{α_f} is:

$$F_{\alpha_f}^{(n,m)} = F_L^{(q_{last})} \tag{3.91}$$

Using the computed volume fraction, the density is updated as follows [7]:

$$\rho^{(n,m)} = \widetilde{\alpha_P}^{(n,m)} \rho_w + (1 - \widetilde{\alpha_P}^{(n,m)}) \rho_a \tag{3.92}$$

with:

$$\widetilde{\alpha_P}^{(n,m)} = \min\left(\max\left(\alpha_P^{(n,m,q_{last})}, 0\right), 1\right) \tag{3.93}$$

At the end of the VOF resolution, with the relaxation zone activated, according to Eq. 2.61, the volume fraction is corrected as follows:

$$\alpha_P^{(n,m)} = (1 - w)\alpha_P^{(n,m,q_{last})} + w\left(\alpha_P^{(n,m\text{-}1)}\right)^{target} \tag{3.94}$$

And finally, the kinetic and dynamic viscosity are calculated:

$$\mu^{(n,m)} = \widetilde{\alpha}_P^{(n,m)} \rho_w + (1 - \widetilde{\alpha}_P^{(n,m)}) \rho_a \tag{3.95a}$$

$$\mu_{eff}^{(n,m)} = \mu^{(n,m)} + \mu_t^{(n,m\text{-}1)} \tag{3.95b}$$

$$\nu^{(n,m)} = \frac{\mu^{(n,m)}}{\rho^{(n,m)}} \tag{3.95c}$$

b) Euler MULES

The MULES algorithm aims to ensure TVD criterion for the spatial schemes even adding the compression term and then does not create new maximum and minimum terms. However, this notion of maximum and minimum is relative to the first estimation $\alpha_P^{(n,m,0)}$ of the volume fraction (Eq. 3.88a). Thanks to the MULES algorithm, the converged result $\alpha_P^{(n,m,q_{last})}$ is bounded by $\alpha_{P_{min}}^{(n,m,0)}$ and $\alpha_{P_{max}}^{(n,m,0)}$. So, $\alpha_P^{(n,m,q_{last})}$ between 0 and 1 is only ensured if $\alpha_P^{(n,m,0)}$

6. The expression of F_{α_f} depends on the selected method
7. This equation is discussed in Sec. 3.3.3

obtained by the predictor step is between 0 and 1. Only the Euler time scheme ensures this property. In the MULES algorithm, some limit values can be imposed but nothing ensures that the algorithm will converge within these limits if $\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}$ does not respect them.

With the Euler implicit scheme the predictor step (3.88) can be written as:

$$\frac{\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}}{\Delta t} + \frac{1}{V_{\mathrm{P}}^{(\mathrm{n,m})}} \sum_{\mathrm{f}} \mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}\right)_{\mathrm{f}(UD)} \left[\phi_{\mathrm{f}}^{(\mathrm{n,m\text{-}1})} - \phi_{\mathrm{mf}}^{(\mathrm{n,m\text{-}1})}\right] = \frac{\alpha_{\mathrm{P}}^{(\mathrm{n\text{-}1})} V_{\mathrm{P}}^{(\mathrm{n\text{-}1})}}{\Delta t\, V_{\mathrm{P}}^{(\mathrm{n,m})}} \tag{3.96a}$$

$$F_{\mathrm{L}}^{(0)} = \mathscr{I}_{\mathrm{UD}}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}\right)_{\mathrm{f}} \left[\phi_{\mathrm{f}}^{(\mathrm{n,m\text{-}1})} - \phi_{\mathrm{mf}}^{(\mathrm{n,m\text{-}1})}\right] \tag{3.96b}$$

and the correction step as:

$$F_{\mathrm{H}}^{(\mathrm{q})} = \mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,q\text{-}1})}\right)_{\mathrm{f}} \left[\phi_{\mathrm{f}}^{(\mathrm{n,m\text{-}1})} - \phi_{\mathrm{mf}}^{(\mathrm{n,m\text{-}1})}\right] + \phi_{\mathrm{comp.f}}^{(\mathrm{n,m,q})} \tag{3.97a}$$

$$F_{\mathrm{L}}^{(\mathrm{q})} = (1 - \lambda^{(\mathrm{q})}) F_{\mathrm{L}}^{(\mathrm{q\text{-}1})} + \lambda^{(\mathrm{q})} F_{\mathrm{H}}^{(\mathrm{q})} \tag{3.97b}$$

$$\alpha^{(\mathrm{n,m,q})} = \alpha^{(\mathrm{n,m,q\text{-}1})} - \frac{\Delta t}{V_{\mathrm{P}}^{(\mathrm{n,m})}} \sum_{\mathrm{f}} \lambda^{(\mathrm{q})} (F_{\mathrm{H}}^{(\mathrm{q})} - F_{\mathrm{L}}^{(\mathrm{q\text{-}1})}) \tag{3.97c}$$

Finally, the volume fraction flux $F_{\alpha_{\mathrm{f}}}$ is respecting the same definition as in the general formulation previously defined (Eq. 3.91).

c) Crank-Nicolson MULES

In OpenFOAM-5 the implicit Crank-Nicolson can be used for the VOF resolution with the MULES algorithm. For the predictor step, the Crank-Nicolson scheme presented in Section 3.2 could be used directly with the general formulation (see in (a)) but it would not ensure $0 \leq \alpha_{\mathrm{P}} \leq 1$. The reason is the contribution of other terms than $\alpha_{\mathrm{P}}^{(\mathrm{n\text{-}1})}$ in the source term. Then, the predictor step is formulated in OpenFOAM-5 with an Euler-like scheme introducing a mixed flux $\Phi_{\mathrm{CN_f}}$:

$$\Phi_{\mathrm{CN_f}}^{(\mathrm{n,m\text{-}1})} = C_{\mathrm{CN}} \left[\phi_{\mathrm{f}}^{(\mathrm{n,m\text{-}1})} - \phi_{\mathrm{mf}}^{(\mathrm{n,m\text{-}1})}\right] + (1 - C_{\mathrm{CN}}) \left[\phi_{\mathrm{f}}^{(\mathrm{n\text{-}1})} - \phi_{\mathrm{mf}}^{(\mathrm{n\text{-}1})}\right]. \tag{3.98}$$

The resulting predictor step is defined as follows:

$$\frac{\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}}{\Delta t} + \frac{1}{V_{\mathrm{P}}^{(\mathrm{n,m})}} \sum_{\mathrm{f}} \mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}\right)_{\mathrm{f}(UD)} \Phi_{\mathrm{CN_f}}^{(\mathrm{n,m\text{-}1})} = \frac{\alpha_{\mathrm{P}}^{(\mathrm{n\text{-}1})} V_{\mathrm{P}}^{(\mathrm{n\text{-}1})}}{\Delta t\, V_{\mathrm{P}}^{(\mathrm{n,m})}} \tag{3.99a}$$

$$F_{\mathrm{L}}^{(0)} = \mathscr{I}_{\mathrm{UD}}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}\right)_{\mathrm{f}} \Phi_{\mathrm{CN_f}}^{(\mathrm{n,m\text{-}1})} \tag{3.99b}$$

This formulation of $F_{\mathrm{L}}^{(0)}$ can be seen as writing the diffusive part obtained with Crank-Nicolson time-scheme:

$$\begin{aligned}\frac{\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}}{\Delta t} &+\frac{1}{V_{\mathrm{P}}^{(\mathrm{n,m})}}\sum_{\mathrm{f}}\left[C_{\mathrm{CN}}\mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}\right)_{\mathrm{f}(UD)}\left[\phi_{\mathrm{f}}^{(\mathrm{n,m\text{-}1})}-\phi_{\mathrm{mf}}^{(\mathrm{n,m\text{-}1})}\right]\right.\\ &\left.+\frac{1}{V_{\mathrm{P}}^{(\mathrm{n,m})}}\sum_{\mathrm{f}}(1-C_{\mathrm{CN}})\mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n\text{-}1})}\right)_{\mathrm{f}(UD)}\left[\phi_{\mathrm{f}}^{(\mathrm{n\text{-}1})}-\phi_{\mathrm{mf}}^{(\mathrm{n\text{-}1})}\right]\right. &=\frac{\alpha_{\mathrm{P}}^{(\mathrm{n\text{-}1})}V_{\mathrm{P}}^{(\mathrm{n\text{-}1})}}{\Delta t\,V_{\mathrm{P}}^{(\mathrm{n,m})}}\end{aligned} \tag{3.100}$$

but making the approximation $\alpha_{\mathrm{P}}^{(\mathrm{n\text{-}1})}\approx\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}$ in the convection term.

The resulting predictor is first-order accurate as would be an Euler one. Consequently, using this predictor, the MULES algorithm aims to reach both spatial and temporal second-order accuracy. The side effect is that second-order time accuracy can be achieved if and only if the second-order spatial accuracy is reached.

Using the Crank-Nicolson time scheme, the targeted high-order flux is formulated as follows:

$$F_{\mathrm{H}}^{(\mathrm{q})}=\underbrace{C_{\mathrm{CN}}\mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,q\text{-}1})}\right)_{\mathrm{f}}\left[\phi_{\mathrm{f}}^{(\mathrm{n,m\text{-}1})}-\phi_{\mathrm{mf}}^{(\mathrm{n,m\text{-}1})}\right]+(1-C_{\mathrm{CN}})F_{\alpha_{\mathrm{f}}}^{(\mathrm{n\text{-}1})}}_{Crank-Nicolson\ formulation}+\phi_{\mathrm{comp.f}}^{(\mathrm{n,m,q})} \tag{3.101}$$

and the correction step:

$$F_{\mathrm{L}}^{(\mathrm{q})}=(1-\lambda^{(\mathrm{q})})F_{\mathrm{L}}^{(\mathrm{q\text{-}1})}+\lambda^{(\mathrm{q})}F_{\mathrm{H}}^{(\mathrm{q})} \tag{3.102a}$$

$$\alpha^{(\mathrm{n,m,q})}=\alpha^{(\mathrm{n,m,q\text{-}1})}-\frac{\Delta t}{V_{\mathrm{P}}^{(\mathrm{n,m})}}\sum_{\mathrm{f}}\lambda^{(\mathrm{q})}(F_{\mathrm{H}}^{(\mathrm{q})}-F_{\mathrm{L}}^{(\mathrm{q\text{-}1})}) \tag{3.102b}$$

Using the off-centering coefficient C_{OC} (Eq. 3.48), $F_{\alpha_{\mathrm{f}}}$ is computed as:

$$F_{\alpha_{\mathrm{f}}}^{(\mathrm{n,m})}=(1+C_{\mathrm{OC}})F_{\mathrm{L}}^{(\mathrm{q_{last}})}-C_{\mathrm{OC}}F_{\alpha_{\mathrm{f}}}^{(\mathrm{n\text{-}1})} \tag{3.103}$$

In Chapter 4, the general MULES formulation keeping a time second-order predictor is tested in spite of the potential loss of TVD behavior. This MULES method only dealing with the spatial discretization is denoted S-MULES in this document.

3.3.3 Discussion on VOF resolution

The previous section details how the VOF algorithm is implemented in *foamStar*. Some parts of this implementation are discussed here.

a) The density-flux expression

As mentioned, one important updated field during the VOF resolution is the density-flux $F_{\rho_f}^{(n,m)}$ (Eq. 3.86). This density flux is the one used for the momentum convection detailed in Section 3.3.4. It implies some discussion on the consistency of its expression.

Time-scheme consistency As mentioned, because of its definition, $F_{\rho_f}^{(n,m)}$ does respect the mass conservation equation (Eq. 3.87). However, the time-scheme coefficients are those imposed by the selected scheme. Consequently, using another time-scheme for the momentum equation would lead to an inconsistent density-flux regarding to the momentum convection and the geometric-conservation law.

Clipping of the volume fraction As detailed, the VOF-MULES algorithm aims to keep $0 \leq \alpha \leq 1$. However, because of the resolution errors, a non-converged MULES step or the use of an non-Euler-like predictor, the updated volume fraction field $\alpha^{(n,m,q_{last})}$ can eventually get some values out of the prescribed interval $[0,1]$. Such values could significantly impair the simulation if they were directly used for ρ and μ calculation. It explains why, in Eq. 3.92, $\rho^{(n,m)}$ is not calculated with ${\alpha_P}^{(n,m,q_{last})}$ but with the clipped volume fraction $\widetilde{\alpha_P}^{(n,m)}$. The consequence of this clipping is that using this formulation of $\rho^{(n,m)}$, the mass conservation equation Eq. 3.87 is not truly verified. Then, if the clipping implies a too large change in the volume fraction, some spurious velocities could occur during the the resolution of the momentum-equation.

b) Predictor step for Crank-Nicolson MULES

With Crank-Nicolson MULES, it can be interesting to question the use of Eq. 3.99 for the predictor step rather than the standard Euler predictor Eq. 3.96. No reference where found in the literature, so looking at the equations, two reasons could explain this choice.

The first one is the respect of the geometric conservation law. Using the currently implemented predictor (Eq. 3.99), the coefficient of the mesh flux is the same as with the Crank-Nicolson scheme. So, the geometric-conservation law is still respected if the field $F_{\rho_f}^{(n,m)}$

is used for the momentum convection with a Crank-Nicolson scheme and a same off-centering coefficient.

The second, and probably the main reason, is the error implied by the formulation of $F_{\alpha_\mathrm{f}}^{(\mathrm{n,m})}$. In the following, the coefficients resulting from the MULES correction at a face f is λ_f and the Crank-Nicolson off-centering coefficient is C_OC. For more simplicity, we assume here a fixed mesh, only one VOF correction loop, no compression term, and a previous time-step solution obtained with a "perfect" MULES correction $\lambda_\mathrm{f}^{(\mathrm{n\text{-}1})} = 1$.

Using $\Phi_{\mathrm{CN_f}}$ as defined in *foamStar*, the resulting $F_{\alpha_\mathrm{f}}^{(\mathrm{n,m})}$ flux at a face f is:

$$\begin{aligned} F_{\alpha_\mathrm{f}}^{(\mathrm{n,m})} &= \phi_\mathrm{f}^{(\mathrm{n,m})} \left[(1-\lambda_\mathrm{f}) \mathscr{I}_\mathrm{UD}\left(\alpha_\mathrm{P}^{(\mathrm{n,m,0})}\right)_\mathrm{f} + \lambda_\mathrm{f} \mathscr{I}\left(\alpha_\mathrm{P}^{(\mathrm{n,m})}\right)_\mathrm{f} \right] \\ &+ C_\mathrm{OC}\, \phi_\mathrm{f}^{(\mathrm{n\text{-}1})} \left[(1-\lambda_\mathrm{f}) \left[\mathscr{I}_\mathrm{UD}\left(\alpha_\mathrm{P}^{(\mathrm{n,m,0})}\right)_\mathrm{f} - \mathscr{I}\left(\alpha_\mathrm{P}^{(\mathrm{n\text{-}1})}\right)_\mathrm{f} \right] \right] \end{aligned} \tag{3.104}$$

As expected, when $\lambda_\mathrm{f} = 1$ the impact of this formulation is:

$$F_{\alpha_\mathrm{f}}^{(\mathrm{n,m})} = \phi_\mathrm{f}^{(\mathrm{n,m})} \mathscr{I}_\mathrm{UD}\left(\alpha_\mathrm{P}^{(\mathrm{n,m})}\right)_\mathrm{f} \tag{3.105}$$

The issue occurs with $\lambda_\mathrm{f} \neq 1$. It implies additional terms in the VOF flux formulation with a contribution of the previous time-step fields $\alpha_\mathrm{P}^{(\mathrm{n\text{-}1})}$ and $\phi_\mathrm{f}^{(\mathrm{n\text{-}1})}$. The undesired contribution is of the order of $|\alpha_\mathrm{P}^{(\mathrm{n,m})} - \alpha_\mathrm{P}^{(\mathrm{n\text{-}1})}|$, so no additional terms are added in regions outside the transition zone.

In the same situation but considering the initialization with the Euler approach, the resulting $F_{\alpha_\mathrm{f}}^{(\mathrm{n,m})}$ flux at a face f is:

$$\begin{aligned} F_{\alpha_\mathrm{f}}^{(\mathrm{n,m})} &= \phi_\mathrm{f}^{(\mathrm{n,m})} \left[(1-\lambda_\mathrm{f}) \mathscr{I}_\mathrm{UD}\left(\alpha_\mathrm{P}^{(\mathrm{n,m,0})}\right)_\mathrm{f} + \lambda_\mathrm{f} \mathscr{I}\left(\alpha_\mathrm{P}^{(\mathrm{n,m})}\right)_\mathrm{f} \right] \\ &+ C_\mathrm{OC}\, \phi_\mathrm{f}^{(\mathrm{n\text{-}1})} \left[(1-\lambda_\mathrm{f}) \left[\mathscr{I}\left(\alpha_\mathrm{P}^{(\mathrm{n\text{-}1})}\right)_\mathrm{f} \right] \right] \end{aligned} \tag{3.106}$$

When $\lambda_\mathrm{f} = 1$ the impact of this formulation is the same as previously. But now, with $\lambda_\mathrm{f} \neq 1$, the contribution of $\alpha_\mathrm{P}^{(\mathrm{n\text{-}1})}$ is not compensated by $\alpha_\mathrm{P}^{(\mathrm{n,m,0})}$. Then, in the wet region ($\alpha > 0$), even with a constant volume fraction, the flux used for the convection term in the momentum equation includes the contributions of $\alpha_\mathrm{P}^{(\mathrm{n\text{-}1})}$ and $\phi_\mathrm{f}^{(\mathrm{n\text{-}1})}$.

In regions where the volume fraction is locally constant (far from the free surface), the MULES coefficient λ_f will not converge to 1 so the exposed situation with $\lambda_\mathrm{f} \neq 1$ concerns a majority of the cell faces of the domain.

Finally, both Euler and Crank-Nicolson approach are first-order predictor, the Crank-Nicolson introducing $\Phi_{\mathrm{CN_f}}$ is the default implementation in OpenFOAM and, as

mentioned, Euler approach can impose strong modifications of the flux even with constant volume fraction. Consequently, only the Crank-Nicolson approach will be considered in the following.

c) The relaxation zone

The corrections induced by the relaxation zone are made at the end of the VOF resolution but only impact the expression of $\alpha^{(\mathrm{n,m})}$, $\mu^{(\mathrm{n,m})}$ and $\nu^{(\mathrm{n,m})}$. The first remark is that the density field ρ and the viscosity field μ are not calculated with the same volume fraction. Consequently, some inconsistencies exist between the volume fraction and the density and density-flux. The second remark is that the numerical volume fraction $\alpha^{(\mathrm{n,m})}$ is not used in the *foamStar* code except in the time scheme applied to the next time-step (n+1) with the source field $\alpha^{(\mathrm{n})} = \alpha^{(\mathrm{n,m_{last}})}$. Consequently, it could be interesting to apply the relaxation zone constraint on the volume fraction only at the end of the time-step rather than inside the PIMPLE loop.

3.3.4 PISO loop: RANSE resolution

At this step, the mesh motion is known and the VOF has been updated. In the moving-domain $\mathscr{R}_{\mathrm{D}}$, the set of equations that are solved in this section is:

$$\nabla \bullet \mathbf{u} = 0 \tag{2.7}$$

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla\bullet(\rho\mathbf{u}\mathbf{u} + (\rho_{\mathrm{w}} - \rho_{\mathrm{a}})\mathbf{u}_{\mathrm{comp}}\mathbf{u}) = -\nabla p - \mathbf{g}\bullet\mathbf{x}\nabla\rho + \nabla\bullet(\mu_{\mathrm{eff}}\nabla\mathbf{u}) + \nabla\mu_{\mathrm{eff}}\bullet\nabla\mathbf{u}^T - \rho\,\mathbf{a}_{\mathrm{cc}_0} \tag{2.29}$$

Table 3.5 indicates the main fields updated during this step.

Table 3.5 – Numerical fields updated by the RANSE resolution

Numerical values	Description
$\mathbf{u}_{\mathrm{P}}^{(\mathrm{n,m})}$	The velocity at cell centers
$\mathbf{u}_{\mathrm{f}}^{(\mathrm{n,m})}$	The velocity at face centers
$\phi_{\mathrm{f}}^{(\mathrm{n,m})}$	The velocity flux at faces
$p_{\mathrm{P}}^{(\mathrm{n,m})}$	The pressure at cell centers

The velocity and pressure variables are computed at the center of cells. So, in order to avoid pressure oscillations solving RANSE with collocated variables, the OpenFOAM algorithm is

using a variant the Rhie and Chow (1983) method in the PISO loop. Figure 3.9 gives a schematic view of the PISO loop. First the loop starts with a predictor step where the velocity field **u** is estimated solving the momentum equation (Eq. 2.29). Then at each iteration of a recursive PISO loop, the pressure field p is updated using both the momentum and continuity equations (Eq. 2.29, Eq. 2.7) followed by the update of the **u** according to the momentum equation. In *foamStar*, the relaxation zone constraint is applied to the velocity field after the last PISO iteration.

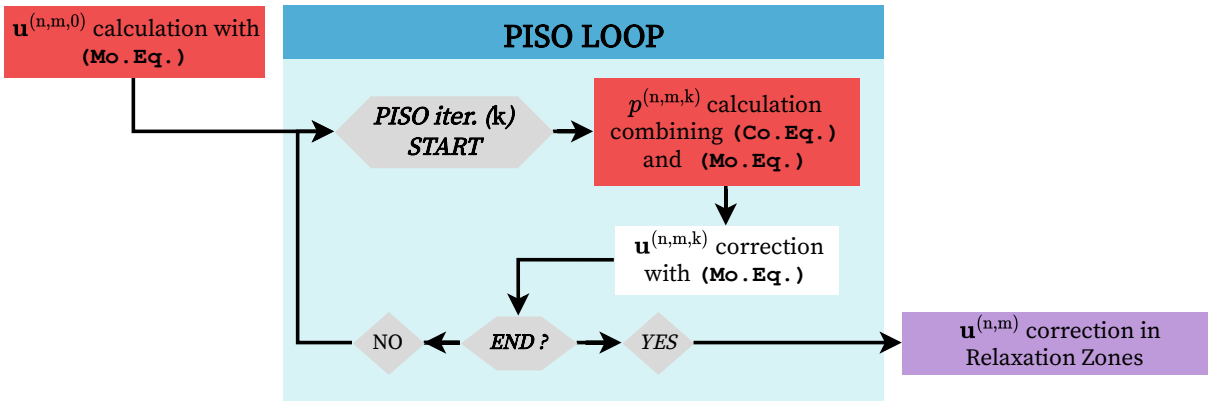


Figure 3.9 – Simplified flowchart of the PISO loop

a) Implicit velocity calculation

The first step is the initialization of the PISO loop with an estimation of the velocity $\mathbf{u}_P^{(n,m,0)}$. To do so the momentum equation 2.29 is discretized as follow,

$$
\begin{aligned}
& c_n^{(n)} \rho_P^{(n,m)} \mathbf{u}_P^{(n,m,0)} \\
& + \frac{1}{V_P^{(n,m)}} \sum_f \left(\mathscr{I}\left(\mathbf{u}_P^{(n,m,0)}\right)_f F_{\rho_f}^{(n,m)} - \mu_{\text{eff}}^{(n,m)} \mathbf{S}_f \bullet \nabla \mathbf{u}|_f^{(n,m,0)} \right) = \\
& \nabla \mathbf{u}|_P^{(n,m-1)} \bullet \nabla \mu_{\text{eff}}|_P^{(n,m)} + \mathscr{I}\left(g z_P^{(n,m)} \nabla \rho|_f^{(n,m)} - \nabla p|_f^{(n,m-1)} \right)_P \\
& - \frac{1}{V_P^{(n,m)}} \left[c_{n-1}^{(n)} (\rho \mathbf{u}_P V_P)^{(n-1)} + c_{n-2}^{(n)} (\rho \mathbf{u}_P V_P)^{(n-2)} + \gamma_{\rho \mathbf{u}}^{(n)} \right] - \rho^{(n,m)} \mathbf{a}_{cc_0}^{(n)}
\end{aligned}
\tag{3.107}
$$

With

$$
\mu_{\text{eff}}^{(n,m)} = \mu^{(n,m)} + \rho^{(n,m)} \nu_t^{(n-1,m)} \tag{3.108}
$$

In this equation only $\mathbf{u}_P^{(n,m,0)}$ is unknown. The resulting linear system is solved using matrix resolution algorithm as mentioned in Section 3.3.7.

b) PISO loop

For this part some new notations have to be introduced in order to decompose the interpolation operator $\mathscr{I}\left(\mathbf{u}_\mathrm{P}\right)_\mathrm{f}$ and gradient operator $\mathbf{S}_\mathrm{f}\bullet\,\boldsymbol{\nabla}\mathbf{u}|_\mathrm{f}$. The decompositions consist in identifying the coefficients $\gamma_{I_\mathrm{f}}^N$, $\gamma_{\Delta_\mathrm{f}}^N$, $\gamma_{I_\mathrm{f}}^P$, $\gamma_{\Delta_\mathrm{f}}^P$, $\gamma_{I_\mathrm{f}}^S$ and $\gamma_{\Delta_\mathrm{f}}^S$ associated to the neighbour cell N, the principal cell P and additional source S relative to the considered operators.

$$\mathscr{I}\left(\mathbf{u}_\mathrm{P}\right)_\mathrm{f} = \gamma_{I_\mathrm{f}}^N\,\mathbf{u}_\mathrm{N} + \gamma_{I_\mathrm{f}}^P\,\mathbf{u}_\mathrm{P} + \gamma_{I_\mathrm{f}}^S \tag{3.109a}$$

$$\mathbf{S}_\mathrm{f}\bullet\,\boldsymbol{\nabla}\mathbf{u}|_\mathrm{f} = \gamma_{\Delta_\mathrm{f}}^N\,\mathbf{u}_\mathrm{N} + \gamma_{\Delta_\mathrm{f}}^P\,\mathbf{u}_\mathrm{P} + \gamma_{\Delta_\mathrm{f}}^S \tag{3.109b}$$

As seen in Section 3.1, these coefficients depend on the cells geometry and the chosen numerical schemes.

In the PISO loop, the momentum equation is decomposed introducing the fields $D_{\mathbf{u}_\mathrm{P}}^{(\mathrm{n,m})}$ and $\boldsymbol{\mathcal{H}}_\mathrm{P}^{(\mathrm{n,m,k})}$. A common way to introduce these fields is to use matrix formulation. Given the matrix system built with Eq. 3.107 applied at each cell center of the numerical domain:

$$\mathbf{M}^{(\mathrm{n,m})}[\mathbf{u}]^{(\mathrm{n,m,0})} = \mathbf{S}^{(\mathrm{n,m})} + \mathbf{G}_\rho^{(\mathrm{n,m})} + \mathbf{G}_p^{(\mathrm{n,m\text{-}1})} \tag{3.110}$$

with $\mathbf{M}^{(\mathrm{n,m})}$ the matrix of the system, $\mathbf{G}_\rho^{(\mathrm{n,m})}$ and $\mathbf{G}_p^{(\mathrm{n,m})}$ the vectors containing the source terms associated with the gradients of density and pressure, respectively, and $\mathbf{S}^{(\mathrm{n,m})}$ the vector containing the other source terms.

Defining $\mathbf{D}^{(\mathrm{n,m})}$ the diagonal matrix from $\mathbf{M}^{(\mathrm{n,m})}$ and $\mathbf{O}^{(\mathrm{n,m})}$ the matrix built with its outer-diagonal coefficients, the matrix system becomes:

$$\mathbf{D}^{(\mathrm{n,m})}[\mathbf{u}]^{(\mathrm{n,m,0})} + \mathbf{O}^{(\mathrm{n,m})}[\mathbf{u}]^{(\mathrm{n,m,0})} = \mathbf{S}^{(\mathrm{n,m})} + \mathbf{G}_\rho^{(\mathrm{n,m})} + \mathbf{G}_p^{(\mathrm{n,m\text{-}1})} \tag{3.111}$$

Then, defining the vector:

$$\mathbf{H}^{(\mathrm{n,m,k})} = -\mathbf{O}^{(\mathrm{n,m})}[\mathbf{u}]^{(\mathrm{n,m,k})} + \mathbf{S}^{(\mathrm{n,m})} \tag{3.112}$$

the following matrix system can be built:

$$\mathbf{D}^{(\mathrm{n,m})}[\mathbf{u}]^{(\mathrm{n,m,0})} = \mathbf{H}^{(\mathrm{n,m,k})} + \mathbf{G}_\rho^{(\mathrm{n,m})} + \mathbf{G}_p^{(\mathrm{n,m,k})} \tag{3.113}$$

For k=0, this matrix system is equivalent to Eq. 3.110.

Finally, $D_{\mathbf{u}_\mathrm{P}}^{(\mathrm{n,m})}$ and $\boldsymbol{\mathcal{H}}_\mathrm{P}^{(\mathrm{n,m,k})}$ are the coefficients of $\mathbf{D}^{(\mathrm{n,m})}$ and $\mathbf{H}^{(\mathrm{n,m,k})}$, and their

expressions are:

$$D_{\mathbf{u}_\mathrm{P}}^{(\mathrm{n,m})} = \frac{1}{V_\mathrm{P}^{(\mathrm{n,m})}} \sum_\mathrm{f} \gamma_{I_\mathrm{f}}^{P} F_{\rho_\mathrm{f}}^{(\mathrm{n,m})} - \frac{\mu_\mathrm{eff}^{(\mathrm{n,m})}}{V_\mathrm{P}^{(\mathrm{n,m})}} \sum_\mathrm{f} \gamma_{I_\mathrm{f}}^{P} + c_\mathrm{n}^{(\mathrm{n})} \rho^{(\mathrm{n,m})} \tag{3.114}$$

$$\begin{aligned} \mathcal{H}_\mathrm{P}^{(\mathrm{n,m,k})} = & -\frac{1}{V_\mathrm{P}^{(\mathrm{n,m})}} \sum_\mathrm{f} \left[\left(\gamma_{I_\mathrm{f}}^{N} \mathbf{u}_\mathrm{N}^{(\mathrm{n,m,k\text{-}1})} + \gamma_{I_\mathrm{f}}^{N} \right) F_{\rho_\mathrm{f}}^{(\mathrm{n,m})} - \mu_\mathrm{eff}^{(\mathrm{n,m})} \left(\gamma_{\Delta_\mathrm{f}}^{N} \mathbf{u}_\mathrm{N}^{(\mathrm{n,m,k\text{-}1})} + \gamma_{\Delta_\mathrm{f}}^{S} \right) \right] \\ & + \boldsymbol{\nabla} \mathbf{u}|_\mathrm{P}^{(\mathrm{n,m\text{-}1})} \cdot \boldsymbol{\nabla} \mu_\mathrm{eff}|_\mathrm{P}^{(\mathrm{n,m})} - \frac{1}{V_\mathrm{P}^{(\mathrm{n,m})}} \left[c_{\mathrm{n\text{-}1}}^{(\mathrm{n})} (\rho \mathbf{u}_\mathrm{P} V_\mathrm{P})^{(\mathrm{n\text{-}1})} + c_{\mathrm{n\text{-}2}}^{(\mathrm{n})} (\rho \mathbf{u}_\mathrm{P} V_\mathrm{P})^{(\mathrm{n\text{-}2})} + \gamma_{\rho \mathbf{u}}^{(\mathrm{n})} \right] \\ & - \rho^{(\mathrm{n,m})} \mathbf{a}_{\mathrm{cc}_0}^{(\mathrm{n})} \end{aligned} \tag{3.115}$$

With such a formulation, the following discretized form of the momentum equation can be written:

$$\mathbf{u}_\mathrm{P}^{(\mathrm{n,m,k})} = \frac{\mathcal{H}_\mathrm{P}^{(\mathrm{n,m,k})}}{D_{\mathbf{u}_\mathrm{P}}^{(\mathrm{n,m})}} + \frac{\mathscr{I}\left(g z_\mathrm{P}^{(\mathrm{n,m})} \boldsymbol{\nabla} \rho|_\mathrm{f}^{(\mathrm{n,m})} - \boldsymbol{\nabla} p|_\mathrm{f}^{(\mathrm{n,m,k})} \right)_\mathrm{P}}{D_{\mathbf{u}_\mathrm{P}}^{(\mathrm{n,m})}} \tag{3.116}$$

To use this formulation, $\boldsymbol{\nabla} p|_\mathrm{f}^{(\mathrm{n,m,k})}$ requires the calculation of $p_\mathrm{P}^{(\mathrm{n,m,k})}$. To do so, ϕ_f representing the velocity flux at face, the following equation is built:

$$\phi_\mathrm{f}^{(\mathrm{n,m,k})} = \left[\mathscr{I}\left(\frac{\mathcal{H}_\mathrm{P}^{(\mathrm{n,m,k})}}{D_{\mathbf{u}_\mathrm{P}}^{(\mathrm{n,m})}} \right)_\mathrm{f} + \frac{g z_\mathrm{f}^{(\mathrm{n,m})} \boldsymbol{\nabla} \rho|_\mathrm{f}^{(\mathrm{n,m})}}{\mathscr{I}\left(D_{\mathbf{u}_\mathrm{P}}^{(\mathrm{n,m})} \right)_\mathrm{f}} - \frac{\boldsymbol{\nabla} p|_\mathrm{f}^{(\mathrm{n,m,k})}}{\mathscr{I}\left(D_{\mathbf{u}_\mathrm{P}}^{(\mathrm{n,m})} \right)_\mathrm{f}} \right] \cdot \mathbf{S}_\mathrm{f} \tag{3.117}$$

On the other hand, using $\phi_\mathrm{f}^{(\mathrm{n,m,k})}$ the continuity equation Eq. 2.7 is discretized as follows

$$\sum_\mathrm{f} \phi_\mathrm{f}^{(\mathrm{n,m,k})} = 0 \tag{3.118}$$

It results in the following equation where the values of $p_\mathrm{P}^{(\mathrm{n,m,k})}$ are the only unknowns:

$$\sum_\mathrm{f} \left[\frac{\boldsymbol{\nabla} p|_\mathrm{f}^{(\mathrm{n,m,k})}}{\mathscr{I}\left(D_{\mathbf{u}_\mathrm{P}}^{(\mathrm{n,m})} \right)_\mathrm{f}} \right] \cdot \mathbf{S}_\mathrm{f} = \sum_\mathrm{f} \left[\mathscr{I}\left(\frac{\mathcal{H}_\mathrm{P}^{(\mathrm{n,m,k})}}{D_{\mathbf{u}_\mathrm{P}}^{(\mathrm{n,m})}} \right)_\mathrm{f} + \frac{g z_\mathrm{f}^{(\mathrm{n,m})} \boldsymbol{\nabla} \rho|_\mathrm{f}^{(\mathrm{n,m})}}{\mathscr{I}\left(D_{\mathbf{u}_\mathrm{P}}^{(\mathrm{n,m})} \right)_\mathrm{f}} \right] \cdot \mathbf{S}_\mathrm{f} \tag{3.119}$$

The cell-center values $p_\mathrm{P}^{(\mathrm{n,m,k})}$ are obtained solving this linear system with the matrix resolution algorithm as defined in Sec. 3.3.7. Rhie and Chow (1983).

Besides, based on the Rhie and Chow (1983) method, the updated pressure field $p_\mathrm{P}^{(\mathrm{n,m,k})}$ is used for the calculation of the velocity-flux $\phi_\mathrm{f}^{(\mathrm{n,m,k})}$ and $\mathbf{u}_\mathrm{P}^{(\mathrm{n,m,k})}$ using Eq. 3.117 and Eq. 3.116. Finally, at the end of a PISO iteration, the quantities $\mathbf{u}_\mathrm{f}^{(\mathrm{n,m,k})}$ are calculated with the following

relation:

$$\mathbf{u}_{\mathrm{f}}^{(\mathrm{n,m,k})} = \underbrace{\frac{\phi_{\mathrm{f}}^{(\mathrm{n,m,k})}}{|\mathbf{S}_{\mathrm{f}}|}\mathbf{n}_{\mathrm{f}}}_{face-normal\ part} + \underbrace{\mathscr{I}\left(\mathbf{u}_{\mathrm{P}}^{(\mathrm{n,m,k})}\right)_{\mathrm{f}} \bullet \left[I - \mathbf{n}_{\mathrm{f}}\mathbf{n}_{\mathrm{f}}\right]}_{tangential\ part} \tag{3.120}$$

With this formulation, the face-normal part of $\mathbf{u}_{\mathrm{f}}^{(\mathrm{n,m,k})}$ verifies the relation $\phi_{\mathrm{f}}^{(\mathrm{n,m,k})} = \mathbf{u}_{\mathrm{f}}^{(\mathrm{n,m,k})} \bullet \mathbf{S}_{\mathrm{f}}$ but the tangential part is the tangential part of the interpolated value of $\mathbf{u}_{\mathrm{P}}^{(\mathrm{n,m,k})}$. The consequence is that in general, $\mathscr{I}\left(\mathbf{u}_{\mathrm{P}}^{(\mathrm{n})}\right)_{\mathrm{f}} \neq \mathbf{u}_{\mathrm{f}}^{(\mathrm{n})}$

After a user-specified number of PISO iteration, the fluid velocity is corrected in the relaxation zone as follows:

$$\mathbf{u}_{\mathrm{P}}^{(\mathrm{n,m})} = (1-w)\mathbf{u}_{\mathrm{P}}^{(\mathrm{n,m,k_{last}})} + w\left(\mathbf{u}_{\mathrm{P}}^{(\mathrm{n,m\text{-}1})}\right)^{\mathrm{target}}, \tag{3.121}$$

and the other fields are defined with the relations,

$$\mathbf{u}_{\mathrm{f}}^{(\mathrm{n,m})} = \mathbf{u}_{\mathrm{f}}^{(\mathrm{n,m,k_{last}})} \tag{3.122a}$$

$$\phi_{\mathrm{f}}^{(\mathrm{n,m})} = \phi_{\mathrm{f}}^{(\mathrm{n,m,k_{last}})} \tag{3.122b}$$

$$p_{\mathrm{P}}^{(\mathrm{n,m})} = p_{\mathrm{P}}^{(\mathrm{n,m,k_{last}})} \tag{3.122c}$$

If the user-specified number of PIMPLE iterations is reached, all the fields except the kinematic viscosity and the turbulent kinetic energy are considered as solved for the time-step n. The following update can be done for each solved field,

$$\psi^{(\mathrm{n})} = \psi^{(\mathrm{n,m_{last}})} \tag{3.123}$$

3.3.5 Turbulence model resolution

At the end of a time-step and before starting the next one, the turbulence model is solved. The turbulence model detailed here is the k-ω model described in Sec. 2.1.3. In the moving reference frame $\mathscr{R}_0$, the two equations evaluated in P are:

$$\frac{\partial \rho k}{\partial t} + \nabla\bullet\rho\mathbf{u}k - \nabla\bullet((\mu + \rho\alpha_k\nu_{\mathrm{t}})\boldsymbol{\nabla}k) = \rho\nu_{\mathrm{t}}p_0 - \frac{2}{3}\rho k\nabla\bullet\mathbf{u} - \rho\beta^* k\omega - \rho P_{\mathrm{b}} \tag{2.24a}$$

$$\frac{\partial \rho\omega}{\partial t} + \nabla\bullet\rho\mathbf{u}\omega - \nabla\bullet((\mu + \rho\alpha_\omega\nu_{\mathrm{t}})\boldsymbol{\nabla}\omega) = \gamma\rho p_0 - \frac{2}{3}\rho\gamma\omega\nabla\bullet\mathbf{u} - \rho\beta^*\omega^2 - \rho(F_1 - 1)CD_{k\omega} \tag{2.24b}$$

with

$$\nu_t = \frac{k}{\omega} \tag{3.124}$$

The updated numerical fields at this step are listed in Table 3.6

Table 3.6 – Numerical fields updated by the turbulence model resolution

Numerical values	Description
$k^{(n)}$	The turbulent kinetic energy at cell centers
$\omega^{(n)}$	The turbulent dissipation rate at cell centers
$\nu_t^{(n)}$	The turbulent viscosity at cell centers

The discretized equations of the turbulence model are:

$$\begin{aligned} &c_n^{(n)} \rho \omega^{(n)} + \frac{1}{V_P^{(n)}} \sum_f \left(\omega^{(n)} F_{\rho_f}^{(n)} - \rho(\nu_w + \alpha_k \nu_t^{(n-1)}) \mathbf{S}_f \bullet \nabla \omega|_f^{(n)} \right) \\ &= \gamma \rho p_0 \frac{\omega^{(n)}}{k^{(n-1)}} - \frac{2}{3} \rho \gamma \omega^{(n)} \nabla \bullet \mathbf{u}|_P^{(n)} - \rho \beta^* \omega^{(n-1)} \omega^{(n)} - \rho(F_1 - 1) C D_{k\omega} - S_{ddt\,\omega}^{(n)} \end{aligned} \tag{3.125}$$

and

$$\begin{aligned} &c_n^{(n)} \rho k^{(n)} + \frac{1}{V_P^{(n)}} \sum_f \left(k^{(n)} F_{\rho_f}^{(n)} - \rho(\nu_w + \alpha_k \nu_t^{(n-1)}) \mathbf{S}_f \bullet \nabla k|_f^{(n)} \right) \\ &= \rho p_0 - \frac{2}{3} \rho k^{(n)} \nabla \bullet \mathbf{u}|_P^{(n)} - \rho \beta^* k^{(n)} \omega^{(n)} - \rho P_b - S_{ddt\,k}^{(n)} \end{aligned} \tag{3.126}$$

The resulting linear matrix systems are solved successively and then the turbulent kinematic viscosity is updated as follows:

$$\nu_t^{(n)} = \frac{a_1 k^{(n)}}{\max\left(a_1 \omega^{(n)}, b_1 F_2 p_0\right)} \tag{3.127}$$

3.3.6 Boundary conditions

Solving numerical equations implies the definition of boundary conditions for the different fields. These boundary conditions are related to the physical model. In this section the boundary conditions (BCs) available in *foamStar* are presented. More information can be found in the OpenFOAM documentation OpenFOAM (2022a) and Greenshields and Weller (2022).

a) Dirichlet boundary conditions

The first category of BCs is the Dirichlet's one. It consists in imposing a value of the field at the boundary. In OpenFOAM, the surface normal gradient of a field ϕ at face f on a boundary with Dirichlet BC is computed as follows:

$$\nabla\phi_{\mathrm{f}} \bullet \mathbf{n}_{\mathrm{f}} = \frac{\phi_{\mathrm{f}} - \phi_{\mathrm{P}}}{\mathbf{PF} \bullet \mathbf{n}_{\mathrm{f}}}. \tag{3.128}$$

Where **PF** is the vector from the cell center P to the face center F and $\mathbf{n}_{\mathrm{f}}$ is the face normal.

In this document the following Dirichlet's BCs provided by OpenFOAM are used:

- `fixedValue`: This BC imposes a user-defined value at the boundary.
- `movingWallVelocity`: This BC is used for the velocity field at walls, it forces the velocity on the wall patch to be equal to the wall velocity obtained solving the body motions.
- `totalPressure`: This BC imposes a user-defined value p_0 minus a dynamic term $p_{\mathrm{BC}} = p_0 - \frac{1}{2}\rho\mathbf{u}^2$
- `kLowReWallFunction`: This BC imposes the value for the turbulent kinetic energy k according to a two-equation turbulence model adapting the boundary condition to the wall distance and fluid velocity.
- `omegaWallFunction`: This BC imposes the value for the turbulent dissipation rate ω according to a two-equation turbulence model.

Some additional boundary conditions are available in `foamStar` [8]:

- `waveVelocity`: It imposes the velocity field at the boundary to be equal to the analytical solution of the user-defined wave-field.
- `waveAlpha`: It imposes the volume fraction at the boundary to be equal to the analytical solution of the user-defined wave-field.

b) Neumann boundary conditions

The second category is the Neumann's one. It consists in imposing the spatial derivation of the field. In this document, the most used Neumann's BC of OpenFOAM is the `zeroGradient`. With this BC, the face value located at a boundary is equal to the center value

8. In OpenFOAM, namesake exists but the implementation is quite different

of the owner cell and the surface normal gradient is equal to zero. Similarly, `fixedGradient` BC allows to impose a certain value for the surface normal gradient on the boundary face.

Two other particular Neumann's BCs are the `fixedFluxPressure` and `fixedFluxExtrapolatedPressure`. These particular BCs are used to impose a consistent surface normal gradient for the pressure field according to RANSE.

c) Special boundary conditions

- `symmetryPlane`: This BC is similar to a `zeroGradient` boundary condition but applying it to a vector field, the component orthogonal to the symmetry plane is removed.
- `cyclic`: This BC is used to link two boundaries as it will be an internal patch. A face with a cyclic boundary condition is connected to two cells similarly to an internal face [9]. The main difference with an internal face is that the neighbour values owned by the linked cell are expressed explicitly during the iterative resolution.
- `processor`: This BC is a particular one as it is not strictly speaking a condition applied to a boundary of the CFD domain. This condition is used to define the boundaries of the sub-domains created for parallel simulation. Parallelizing the computation, each sub-domain is affected to one CPU. This `processor` BC is located at the boundary between two sub-domains linking the values between cells sharing the same face. It allows the transfer of values between processors during the resolution of the equations. As for `cyclic` BC, the behaviour of a face with `processor` BC is similar to the one of an internal cell but with the explicit formulation of neighbours values.
- `calculated`: With this BC, the value of the fields are calculated during the simulation.

3.3.7 Matrix equation resolution

For a domain with N cells, combination of linear equations involving unknowns ψ_i from several adjacent cells (equations with red terms) result in a N-by-N matrix system with the following form:

$$\mathbf{A}\boldsymbol{\Psi} = \mathbf{S}, \quad \text{with} \quad \boldsymbol{\Psi} = [\psi_i]_{i=1}^{N} \tag{3.129}$$

9. An other BC is available for meshes with non-matching boundary mesh requiring interpolations (`CyclicAMI`)

This matrix system can be solved with different algorithms whose efficiency depends on the structure of $\mathbf{A}$ that is directly linked to the equation type. For instance, a Poisson equation leads to other matrix characteristics than a pure convection equation. In the literature a large panel of solvers can be found and some are commonly used for CFD equations. Overviews of the different solvers can be found in Ferziger and Peric (2012) or more specifically for OpenFOAM in Greenshields and Weller (2022). The linear solvers used in this document are:

— `smoothSolver` with `symGaussSeidel`: This type of solver is based on the Gauss-Seidel method.

— `PBiCGStab`: This is a preconditioned Bi-Conjugate Gradient Stabilized method (Van der Vorst, 1992). It is a commonly used in CFD as a robust method for asymmetric matrix system resolution.

— `GAMG`: It is a Geometric Algebraic Multi-Grid solver. This method can potentially provide faster resolution than others but it will depend on the chosen controlled parameter. (Jemcov et al., 2007; Jasak et al., 2007a)

Residuals Any of the mentioned linear solver is based on iterative algorithms. Conditions have to be specified to stop the algorithm when the error of the solution is deemed to be satisfactory. These conditions are named "tolerance" and the error measurement is done defining the "residual".

For an equation $\mathbf{A}\Psi = \mathbf{S}$, at the q$^{\text{th}}$ iteration of the linear solver, the OpenFOAM formulation of the residual vector $\mathbf{R}^{(\mathrm{q})}$ and the normalized residual $r^{(\mathrm{q})}$ is as follows:

$$\mathbf{R}^{(\mathrm{q})} = \mathbf{S} - \mathbf{A}\Psi^{(\mathrm{q})} \tag{3.130a}$$

$$n = \sum_{i=1}^{N} \left(\left| \left(\mathbf{A}\Psi^{(0)} \right)_i - \left(\mathbf{A}\overline{\Psi}^{(0)} \right)_i \right| + \left| \mathbf{S}_i - \left(\mathbf{A}\overline{\Psi}^{(0)} \right)_i \right| \right) \tag{3.130b}$$

$$r^{(\mathrm{q})} = \frac{1}{n} \sum_{i=1}^{N} \left| \mathbf{R}_i^{(\mathrm{q})} \right| \tag{3.130c}$$

Where $\overline{\Psi}^{(0)}$ is the average of the solution. Its components are defined by:

$$\forall i \in [|1, N|], \quad \overline{\Psi}_i^{(0)} = \frac{1}{N} \sum_{j=1}^{N} \Psi_j^{(0)} \tag{3.131}$$

It is a measure of the distance between the exact solution of the matrix system Ψ and the approximated one $\Psi^{(\mathrm{q})}$. With the OpenFOAM's formulation, the normalized residual $r^{(\mathrm{q})}$

is normalised by a quantity n relative to the initial solution $\mathbf{\Psi}^{(0)}$ used in the linear solver loop. This normalized residual is kept between 0 and 1 and its value depends on the initial solution. A particular case is the one with a uniform initial solution, $\mathbf{\Psi}^{(0)} = \overline{\mathbf{\Psi}}^{(0)}$, matching with the approximated solution $\mathbf{\Psi}^{(\mathrm{q})}$. In this situation the normalized residual is equal to 1. Then, if in addition these quantities are equal to the exact solution $\mathbf{\Psi}$, $n = 0$, $\mathbf{R}^{(\mathrm{q})} = \mathbf{0}$. To remain consistent and avoiding to divide by 0 thanks to limiters, the output normalized residual remains $r^{(\mathrm{q})} = 1$.

Using the normalized residual formulation, two tolerance parameters are available in order to control the stopping condition of the linear solver. The first one is the absolute tolerance ρ_{err} and the second one is the relative tolerance $\varepsilon_{\mathrm{err}}$. With these quantities, and an additional limit number of iterations q_{lim}, the linear solver ends after the $q_{\mathrm{last}}^{\mathrm{th}}$ step if,

$$r^{(\mathrm{q_{last}})} < \rho_{\mathrm{err}} \quad \text{or} \quad \frac{r^{(\mathrm{q_{last}})}}{r^{(0)}} < \varepsilon_{\mathrm{err}} \quad \text{or} \quad q_{\mathrm{last}} = q_{\mathrm{lim}} \tag{3.132}$$

Thus, the approximated solution resulting from the linear solver is $\mathbf{\Psi}^{(\mathrm{q_{last}})}$.

In the following, an alternative notation $Res\left(\mathbf{A}, \mathbf{S}, \psi^{(\mathrm{q})}\right) = r^{(\mathrm{q})}$ can be used in order to clarify the matrix system and approximated solution used to compute the normalized residual.

3.3.8 Flow chart

In the previous sections, all the steps of the resolution process have been detailed. It is now possible to build a flow-chart representing all the *foamStar* algorithm with more details than the one presented in Figure 3.4. The structure of the flow-chart shown in Figure 3.10 is as built as follows:

Colorized loop-boxes Four main loops were previously identified:

- — The green Time-Loop that is the global loop of the algorithm;
- — The yellow PIMPLE-Loop described along this current section (3.3);
- — The blue VOF-Loop described in Section 3.3.2
- — The purple PISO-Loop described in Section 3.3.4

White boxes with colored edges These boxes contain the numerical recursive variables from the previous iteration of the corresponding Loop (with same color).

Diamond shape START boxes These boxes indicate the starting point of a given loop. The list of initial values used by the Loop (index 0) are connected to the corresponding box.

{END, YES, NO} structures They correspond to steps verifying the end condition of a given loop. The variables connected to the **YES** box are the variables updated by the loop.

Colored boxes with text Those boxes represent a sub-algorithm used for some variables update. The corresponding algorithm is detailed in the previous sections. The red boxes correspond to the use of linear solver for matrix system resolution. In previous sections, the equations related to these boxes are identifiable by the red color of the updated variable.

White boxes with thin black edge These boxes only contain one variable and correspond to the update of it. In previous sections, the equations related to this variable update are identifiable by the red color of the updated variable.

Solid arrows These arrows are the structure of the algorithm and are used to identify the sequence.

Dashed arrows These arrow identify the use of some variables inside equations. When the arrow is connected to a loop-box it means that the variable is used in almost all the equations of the loop.

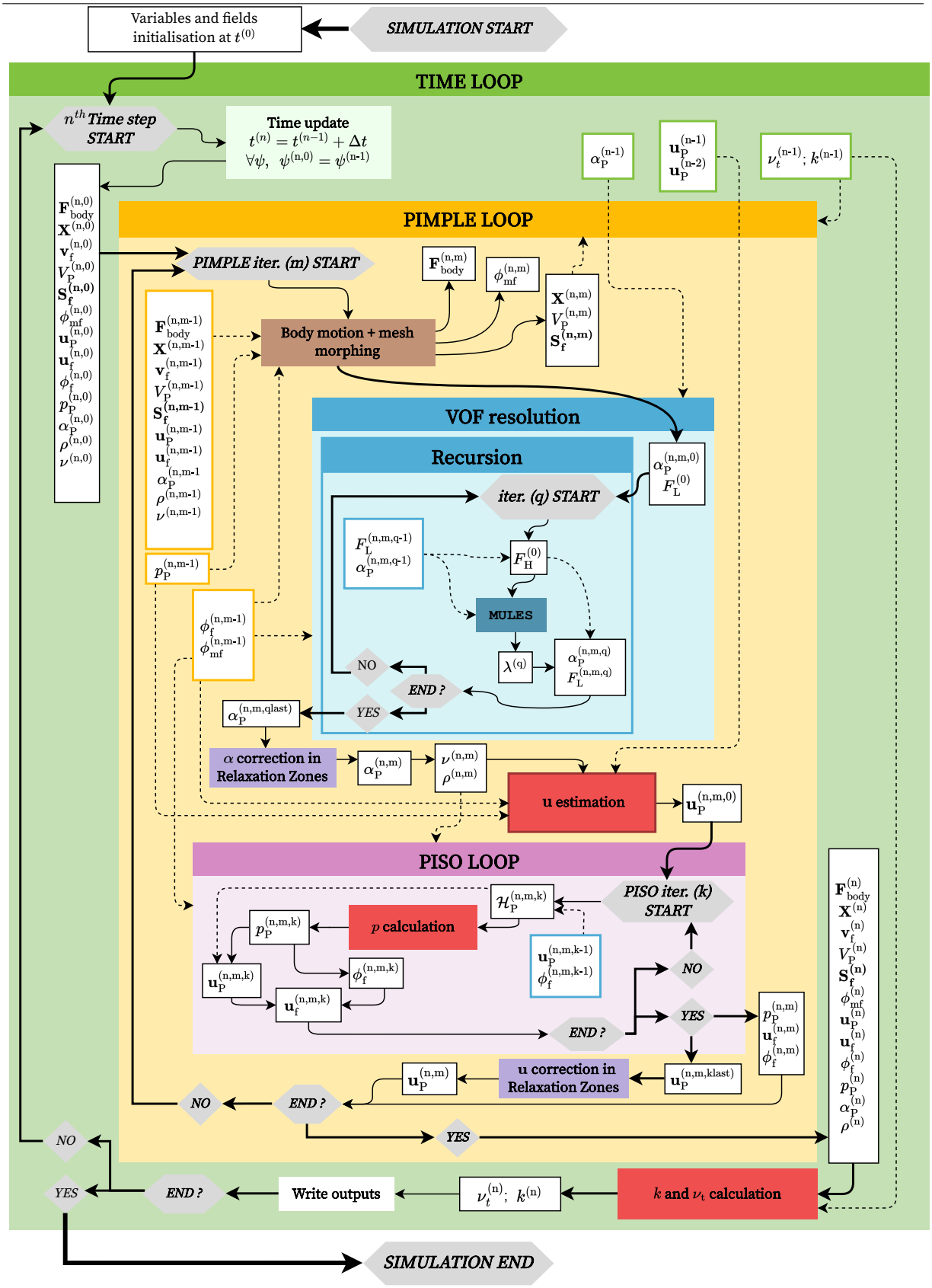


Figure 3.10 – *foamStar* global flowchart

3.4 Iterative error

In the previous sections of this chapter the system of equations and the iterative algorithms have been defined. In order to control the quality of the resolution it is needed to define a metric reflecting the error between the approximated solutions of the discretized equations and their exact values. This error is commonly named the "iterative error". The choice of the error estimator has been discussed in Jasak (1996). With OpenFOAM the convenient error estimator for unsteady problem resolution is based on the normalized residual calculation exposed in Section 3.3.7. As previously mentioned, in *foamStar* matrix systems are used to solve the VOF convection, the momentum predictor, the pressure equation and the two equations of the turbulence model. Due to the PIMPLE and PISO algorithms (Sec. 3.3.4), successive matrix resolutions are made. As the equations are coupled, at a given time-step n and PIMPLE iteration m, the update of the solution $\psi_a^{(\mathrm{n,m})}$ solving a matrix system $\left\{\mathbf{A}_a^{(\mathrm{n,m})},\ \mathbf{S}_a^{(\mathrm{n,m})}\right\}$ implies the creation of a new matrix system $\left\{\mathbf{A}_b^{(\mathrm{n,m})},\ \mathbf{S}_b^{(\mathrm{n,m})}\right\}$ for the field ψ_b with $\left\{\mathbf{A}_b^{(\mathrm{n,m})},\mathbf{S}_b^{(\mathrm{n,m})}\right\} \neq \left\{\mathbf{A}_b^{(\mathrm{n,m-1})},\mathbf{S}_b^{(\mathrm{n,m-1})}\right\}$. Consequently the initial normalized residual for the matrix system b:

$$r_{\psi_b}^{(\mathrm{n,m,0})} = Res\left(\mathbf{A}_b^{(\mathrm{n,m})},\mathbf{S_b}^{(\mathrm{n,m})},\psi_b^{(\mathrm{n,m,0})}\right) = Res\left(\mathbf{A}_b^{(\mathrm{n,m})},\mathbf{S_b}^{(\mathrm{n,m})},\psi_b^{(\mathrm{n,m\text{-}1,q_{last}})}\right)$$

can be used to quantify the distance between the two matrix systems $\left\{\mathbf{A}_b^{(\mathrm{n,m})},\mathbf{S}_b^{(\mathrm{n,m})}\right\}$ and $\left\{\mathbf{A}_b^{(\mathrm{n,m-1})},\mathbf{S}_b^{(\mathrm{n,m-1})}\right\}$ and then for the estimation of the iterative error (Jasak, 1996).

3.4.1 Iterative error in V&V procedure

V&V procedure

The Verification consists of verifying the code implementation. A solver is verified when, reducing the discretization step, the numerical uncertainty evolves consistently with the numerical schemes which were implemented and the expected mathematical solution. On the other part, the validation consists of verifying the agreement between the mathematical model and the experimental results. When the exact mathematical solution is not analytically reachable. Thus, the validation has to be discussed relatively to the numerical and experimental uncertainty. In the following, a simplified presentation of a standard validation process (McHale et al., 2009) is set out.

For the simulation of a physical problem, if n numerical results $(\Phi_i)_1^n$ are obtained,

the general concept of V&V is to evaluate the numerical and the experimental uncertainties respectively noted $U_{\text{CFD,i}}$ and U_{EFD} and to compare them to the experimental results Φ_{EFD}. Defining $E_i = \Phi_i - \Phi_{\text{EFD}}$ the error between the CFD and EFD results, the validation is assumed achieved if $U_{\text{CFD,i}} + U_{\text{EFD}} > |E_i|$.

The problem stays in the fact that the validation assessment depends on the quantification method of $U_{\text{CFD,i}}$ and U_{EFD}. The quantification method of the uncertainty $U_{\text{CFD,i}}$ depends on the numerical results sampling (solver implementations, inputs, discretizations ...). The procedure proposed by Eça and Hoekstra (2014) is well adapted to a single quantity verification, for instance, a drag force. For multiple quantities, for instance, series of probes, multivariate-metric approaches exist. An example is the one proposed by Hills (2006) and used by Pereira et al. (2017).

As mentioned in Stern et al. (2017) and Roache (2019), some bias can exist with the Validation process. In fact, only considering the criterion $U_{\text{CFD,i}} + U_{\text{EFD}} > |E_i|$, small uncertainties can lead to $U_{\text{CFD,i}} + U_{\text{EFD}} << |E_i|$, meaning that the numerical and mathematical models fail to model the physics with a modeling error $\delta_{\text{M}} \approx |E_i|$. However, this modeling error can be small enough to reach the expectations of any industrial or research team. In practice, for specific applications, it can be more consistent to define a target error δ_{T} such as the models are still assumed valid if $U_{\text{CFD,i}} + U_{\text{EFD}} < |E_i|$ but $\delta_{\text{T}} > |E_i|$. On the other side, with such Validation criterion a large uncertainty would easily lead to $U_{\text{CFD,i}} + U_{\text{EFD}} > |E_i|$. It means that the coarser the simulations, the more valid they are. The reason is the term "valid" can be understood as follows: *For the application in question, the mathematical and numerical models are not invalidated by the numerical test cases and the method used to quantify uncertainties. The modeling error δ_M cannot be estimated.* For industrial processes, it is needed to compare the benefit of discretizing more with the induced additional computational cost. Some numerical models potentially have a better ratio between modeling errors and computational cost than others. Consequently, for the present work, the analyses of solver performances have to remain consistent with the scope of application of *foamStar* and the accuracy expected by the naval industry. Discussion of this question can be found in Larsson et al. (2013).

Iterative error consideration

As detailed in Eça and Hoekstra (2013), the Verification relies on the evaluation of the numerical error that can be decomposed into three different types:

Round-off error This error is due to the zero machine and is deeply negligible for RANSE-CFD naval application.

Iterative error As mentioned above, this error is the error between the approximated solutions of the discretized equations and their exact values.

Discretisation error This error is the error between the exact solution of the discretized system and the analytical solution of the mathematical model.

For common Verification procedure, the numerical error estimation is based on the study of the convergence of the numerical solver according to the discretization level. As the accessible data is not the exact solution of the discretized system but the approximated one, it is commonly required to get a negligible iterative error regarding to the total numerical error so that the numerical error is dominated by the discretization error (Larsson et al., 2013).

3.5 Numerical solver modifications and new implementations

3.5.1 Backward MULES

As explained, in OpenFOAM, the VOF propagation allows the use of Crank-Nicolson scheme with the MULES algorithm. The implementation of the Crank-Nicolson MULES consists in using the same method as for Euler MULES but defining differently the initial flux and the corrected flux. The initial flux is built introducing a mixed flux $\Phi_{\mathrm{CN_f}}$ based on the idea of considering $\alpha_{\mathrm{P}}^{(\mathrm{n\text{-}1})} \approx \alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}$ in the discretized equation with Crank-Nicolson time-scheme (see Sec. 3.3.2).

With the current OpenFOAM implementation, the second-order implicit backward scheme (named "backward" in this document for simplicity) is available. However, it cannot be used for the VOF resolution with MULES. So, the only second-order implicit scheme with MULES is the Crank-Nicolson one. Using Crank-Nicolson for the VOF resolution while Backard is used for the momentum equation, causes some inconsistencies between the mass flux obtained during the VOF resolution and its use in the momentum equation (see Section 3.3.3 for more details). To fix this, some modifications were done in order to use in order to use backward formulation with the MULES algorithm.

As mentioned in Section 3.3.2, the MULES algorithm is based on the flux correction method, starting the resolution with a predictor step and an initial low-order flux that would be iteratively corrected using an high-order flux as a target. As mentioned, with the Crank-Nicolson MULES those low-order and high-order blendings can be done with both spatial and temporal

schemes.

As for the Crank-Nicolson scheme, using the MULES algorithm only for the spatial accuracy and keeping a second-order accurate time scheme is possible. However, by doing so, the predictor step will not ensure $0 \leq \alpha_{\mathrm{P}} \leq 1$.

As detailed in Section 3.2, for the VOF equation resolution the Crank-Nicolson formulation differs from an implicit Euler formulation because of an explicit contribution of the volume-fraction flux of the previous time-step $F_{\alpha_{\mathrm{f}}}^{(\mathrm{n\text{-}1})}$. On the other side, the backward formulation differs from Euler because of a contribution of the cell-center volume fraction of the penultimate computed time-step $\alpha_{\mathrm{P}}^{(\mathrm{n\text{-}2})}$. The problem is that this additional source term is not a flux and then cannot be treated directly by the MULES algorithm [10]. Then the idea is to express this cell-center source term as a flux and apply an analog method as the one used for the Crank-Nicolson algorithm.

To express a flux equivalent to the standard backward formulation the following discretized VOF equation is assumed to be verified:

$$c_{\mathrm{n}}^{(\mathrm{n})}\alpha_{\mathrm{P}}^{(\mathrm{n})}V_{\mathrm{P}}^{(\mathrm{n})} + c_{\mathrm{n\text{-}1}}^{(\mathrm{n})}\alpha_{\mathrm{P}}^{(\mathrm{n\text{-}1})}V_{\mathrm{P}}^{(\mathrm{n\text{-}1})} + c_{\mathrm{n\text{-}2}}^{(\mathrm{n})}\alpha_{\mathrm{P}}^{(\mathrm{n\text{-}2})}V_{\mathrm{P}}^{(\mathrm{n\text{-}2})} + \sum_f F_{\alpha_{\mathrm{f}}}^{(\mathrm{n})} = 0 \tag{3.133}$$

with

$$c_{\mathrm{n}}^{(\mathrm{n})} := \frac{1}{\Delta t_{\mathrm{n}}} + \frac{1}{\Delta t_{\mathrm{n}} + \Delta t_{\mathrm{n\text{-}1}}} \tag{3.134a}$$

$$c_{\mathrm{n\text{-}1}}^{(\mathrm{n})} := -\frac{\Delta t_{\mathrm{n}} + \Delta t_{\mathrm{n\text{-}1}}}{\Delta t_{\mathrm{n}}\Delta t_{\mathrm{n\text{-}1}}} \tag{3.134b}$$

$$c_{\mathrm{n\text{-}2}}^{(\mathrm{n})} := \frac{\Delta t_{\mathrm{n}}}{\Delta t_{\mathrm{n\text{-}1}}(\Delta t_{\mathrm{n}} + \Delta t_{\mathrm{n\text{-}1}})} \tag{3.134c}$$

As $c_{\mathrm{n}}^{(\mathrm{n})} + c_{\mathrm{n\text{-}1}}^{(\mathrm{n})} + c_{\mathrm{n\text{-}2}}^{(\mathrm{n})} = 0$, this equation can be rearranged as follows:

$$c_{\mathrm{n}}^{(\mathrm{n})}\left(\alpha_{\mathrm{P}}^{(\mathrm{n})}V_{\mathrm{P}}^{(\mathrm{n})} - \alpha_{\mathrm{P}}^{(\mathrm{n\text{-}1})}V_{\mathrm{P}}^{(\mathrm{n\text{-}1})}\right) - c_{\mathrm{n\text{-}2}}^{(\mathrm{n})}\left(\alpha_{\mathrm{P}}^{(\mathrm{n\text{-}1})}V_{\mathrm{P}}^{(\mathrm{n\text{-}1})} - \alpha_{\mathrm{P}}^{(\mathrm{n\text{-}2})}V_{\mathrm{P}}^{(\mathrm{n\text{-}2})}\right) + \sum_f F_{\alpha_{\mathrm{f}}}^{(\mathrm{n})} = 0 \tag{3.135}$$

10. An option exists in order to add such source terms in the MULES algorithm. However, it will not ensure $0 \leq \alpha_{\mathrm{P}} \leq 1$

Then, for a face f, introducing $F^{(\mathrm{n})}_{\mathrm{bck_f}}$ recursively defined as:

$$F^{(0)}_{\mathrm{bck_f}} = 0 \tag{3.136a}$$

$$F^{(\mathrm{n})}_{\mathrm{bck_f}} = \frac{c^{(\mathrm{n})}_{\mathrm{n\text{-}2}}}{c^{(\mathrm{n\text{-}1})}_{\mathrm{n}}} \left(F^{(\mathrm{n\text{-}1})}_{\alpha_\mathrm{f}} + F^{(\mathrm{n\text{-}1})}_{\mathrm{bck_f}} \right) \tag{3.136b}$$

the discretized VOF equation becomes:

$$c^{(\mathrm{n})}_{\mathrm{n}} \alpha^{(\mathrm{n})}_{\mathrm{P}} + \frac{1}{V^{(\mathrm{n})}_{\mathrm{P}}} \sum_f F^{(\mathrm{n})}_{\alpha_\mathrm{f}} + \frac{1}{V^{(\mathrm{n})}_{\mathrm{P}}} \sum_f F^{(\mathrm{n})}_{\mathrm{bck_f}} = \frac{c_\mathrm{n} \alpha^{(\mathrm{n\text{-}1})}_{\mathrm{P}} V^{(\mathrm{n\text{-}1})}_{\mathrm{P}}}{V^{(\mathrm{n})}_{\mathrm{P}}} \tag{3.137}$$

For more visibility the coefficient C_{BCK} is introduced as follow:

$$C_{\mathrm{BCK}} = \frac{1}{c_\mathrm{n} \Delta t} \tag{3.138a}$$

$$\frac{\alpha^{(\mathrm{n})}_{\mathrm{P}}}{\Delta t} + \frac{C_{\mathrm{BCK}}}{V^{(\mathrm{n})}_{\mathrm{P}}} \left[\sum_f F^{(\mathrm{n})}_{\alpha_\mathrm{f}} + \sum_f F^{(\mathrm{n})}_{\mathrm{bck_f}} \right] = \frac{\alpha^{(\mathrm{n\text{-}1})}_{\mathrm{P}} V^{(\mathrm{n\text{-}1})}_{\mathrm{P}}}{\Delta t \, V^{(\mathrm{n})}_{\mathrm{P}}} \tag{3.138b}$$

It is important to mention that Eq. 3.133 is not really at the end of a time-step because of the approximation of the matrix resolution and additional corrections (relaxation zone, volume-fraction clipping ...) that are not introduced in $F^{(\mathrm{n})}_{\alpha_\mathrm{f}}$. Consequently, using the presented backward-Flux formulation is not numerically equivalent to the standard one. However, this error exists despite the use of the presented formulation.

Thanks to this formulation, the backward MULES can be implemented as it was done for Crank-Nicolson MULES but with C_{BCK} coefficient instead of C_{CN} and $(1 - C_{\mathrm{BCK}})$ and $F^{(\mathrm{n})}_{\mathrm{bck_f}}$ instead of $F^{(\mathrm{n\text{-}1})}_{\alpha_\mathrm{f}}$.

For the following, the introduction of a recursive flux $\phi_{\mathrm{bck_f}}$ is needed:

$$\phi^{(0)}_{\mathrm{bck_f}} = 0 \tag{3.139a}$$

$$\phi^{(\mathrm{n})}_{\mathrm{bck_f}} = \frac{c^{(\mathrm{n})}_{\mathrm{n\text{-}2}}}{c^{(\mathrm{n\text{-}1})}_{\mathrm{n}}} \left(\phi^{(\mathrm{n\text{-}1})}_{\mathrm{f}} - \phi^{(\mathrm{n\text{-}1})}_{\mathrm{mf}} + \phi^{(\mathrm{n\text{-}1})}_{\mathrm{bck_f}} \right) \tag{3.139b}$$

As for the Crank-Nicolson MULES, in order to ensure $0 \leq \alpha_\mathrm{P} \leq 1$, a mixed flux $\Phi_{\mathrm{BCK_f}}$ is introduced in the predictor step:

$$\Phi^{(\mathrm{n,m\text{-}1})}_{\mathrm{BCK_f}} = C_{\mathrm{BCK}} \left[\phi^{(\mathrm{n,m\text{-}1})}_{\mathrm{f}} - \phi^{(\mathrm{n,m\text{-}1})}_{\mathrm{mf}} \right] + C_{\mathrm{BCK}} \, \phi^{(\mathrm{n})}_{\mathrm{bck_f}} \tag{3.140}$$

The resulting predictor step is as follows:

$$\frac{\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}}{\Delta t}+\frac{1}{V_{\mathrm{P}}^{(\mathrm{n,m})}}\sum_{f}\mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}\right)_{\mathrm{f}(UD)}\Phi_{\mathrm{BCK_f}}^{(\mathrm{n,m\text{-}1})}=-\frac{\alpha_{\mathrm{P}}^{(\mathrm{n\text{-}1})}V_{\mathrm{P}}^{(\mathrm{n\text{-}1})}}{\Delta t\,V_{\mathrm{P}}^{(\mathrm{n,m})}} \tag{3.141a}$$

$$F_{\mathrm{L}}^{(0)}=\mathscr{I}_{\mathrm{UD}}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}\right)_{\mathrm{f}}\Phi_{\mathrm{BCK_f}}^{(\mathrm{n,m\text{-}1})} \tag{3.141b}$$

The backward MULES predictor is built using the recursive flux $\phi_{\mathrm{bck_f}}$ whereas the Crank-Nicolson MULES predictor is created using only the face flux and mesh flux at the previous time-step. Then, for Crank-Nicolson MULES, it is correct to consider the error $\left\|\left(\phi_{\mathrm{f}}^{(\mathrm{n\text{-}1})}-\phi_{\mathrm{mf}}^{(\mathrm{n\text{-}1})}\right)\mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}\right)_{\mathrm{f}}-F_{\alpha_{\mathrm{f}}}^{(\mathrm{n\text{-}1})}\right\|=O(\Delta t)$ directly whereas for backward MULES it is necessary to verify the order of the error $\left\|\phi_{\mathrm{bck_f}}^{(\mathrm{n})}\mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}\right)_{\mathrm{f}}-F_{\alpha_{\mathrm{f}}}^{(\mathrm{n\text{-}1})}\right\|$. The verification of $\left\|\phi_{\mathrm{bck_f}}^{(\mathrm{n})}\mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}\right)_{\mathrm{f}}-F_{\alpha_{\mathrm{f}}}^{(\mathrm{n\text{-}1})}\right\|=O(\Delta t)$ is done at the end of this section.

Finally, the resulting predictor is first-order accurate as for Crank-Nicolson MULES.

Using the backward-Flux formulation, the targeted high order flux is formulated as follows:

$$F_{\mathrm{H}}^{(\mathrm{q})}=\underbrace{C_{\mathrm{BCK}}\mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,q\text{-}1})}\right)_{\mathrm{f}}\left[\phi_{\mathrm{f}}^{(\mathrm{n,m\text{-}1})}-\phi_{\mathrm{mf}}^{(\mathrm{n,m\text{-}1})}\right]+C_{\mathrm{BCK}}F_{\mathrm{bck_f}}^{(\mathrm{n})}}_{Backward-Flux\ formulation}+\underbrace{\phi_{\mathrm{comp.f}}^{(\mathrm{n,m,q})}}_{Comp.\ term} \tag{3.142}$$

and the correction step remains the same:

$$F_{\mathrm{L}}^{(\mathrm{q})}=(1-\lambda^{(\mathrm{q})})F_{\mathrm{L}}^{(\mathrm{q\text{-}1})}+\lambda^{(\mathrm{q})}F_{\mathrm{H}}^{(\mathrm{q})} \tag{3.143a}$$

$$\alpha^{(\mathrm{n,m,q})}=\alpha^{(\mathrm{n,m,q\text{-}1})}-\frac{\Delta t}{V_{\mathrm{P}}^{(\mathrm{n,m})}}\sum_{f}\lambda^{(\mathrm{q})}(F_{\mathrm{H}}^{(\mathrm{q})}-F_{\mathrm{L}}^{(\mathrm{q\text{-}1})}) \tag{3.143b}$$

Then, $F_{\alpha_{\mathrm{f}}}$ is computed as:

$$F_{\alpha_{\mathrm{f}}}^{(\mathrm{n,m})}=\frac{1}{C_{\mathrm{BCK}}}F_{\mathrm{L}}^{(\mathrm{q_{last}})}-F_{\mathrm{bck_f}}^{(\mathrm{n})} \tag{3.144}$$

Finally, thanks to this implementation, using $F_{\rho_{\mathrm{f}}}^{(\mathrm{n,m})}$ definition introduced in Section 3.3.2 and renamed here:

$$F_{\rho_{\mathrm{f}}}^{(\mathrm{n,m})}=(\rho_{\mathrm{w}}-\rho_{\mathrm{a}})F_{\alpha_{\mathrm{f}}}^{(\mathrm{n,m})}+\rho_{\mathrm{a}}\left[\phi_{\mathrm{f}}^{(\mathrm{n,m\text{-}1})}+\phi_{\mathrm{mf}}^{(\mathrm{n,m\text{-}1})}\right], \tag{3.86}$$

the discretized form of the mass conservation with Backard scheme is ensured [11].

11. The same discussion on alpha clipping as it is done with Crank-Nicolson in Sec. 3.3.2 can be done here.

Verification of the predictor error Consider that for $n \in \mathbb{I}^{+*}$ and for all $a \in [|0, n-1|]$, the following relation is verified [12]:

$$\left\| \left[\phi_{\mathrm{f}}^{(\mathrm{a})} - \phi_{\mathrm{mf}}^{(\mathrm{a})} \right] \mathscr{I} \left(\alpha_{\mathrm{P}}^{(\mathrm{a})} \right)_{\mathrm{f}} - F_{\alpha_{\mathrm{f}}}^{(\mathrm{a})} \right\| = O(\Delta t)$$

Besides, because of the Taylor-Young theorem:

$$\left\| \left[\phi_{\mathrm{f}}^{(\mathrm{a})} - \phi_{\mathrm{mf}}^{(\mathrm{a})} \right] \mathscr{I} \left(\alpha_{\mathrm{P}}^{(\mathrm{a+1})} \right)_{\mathrm{f}} - \left[\phi_{\mathrm{f}}^{(\mathrm{a})} - \phi_{\mathrm{mf}}^{(\mathrm{a})} \right] \mathscr{I} \left(\alpha_{\mathrm{P}}^{(\mathrm{a})} \right)_{\mathrm{f}} \right\| = O(\Delta t)$$

So, there is $M \in \mathbb{R}^*$ such that, for all $a \in [|0, n-1|]$, the volume-fraction flux verifies:

$$\left\| \left[\phi_{\mathrm{f}}^{(\mathrm{a})} - \phi_{\mathrm{mf}}^{(\mathrm{a})} \right] \mathscr{I} \left(\alpha_{\mathrm{P}}^{(\mathrm{a+1})} \right)_{\mathrm{f}} - F_{\alpha_{\mathrm{f}}}^{(\mathrm{a})} \right\| \leq M \Delta t$$

and

$$\left\| \left[\phi_{\mathrm{f}}^{(\mathrm{a})} - \phi_{\mathrm{mf}}^{(\mathrm{a})} \right] \mathscr{I} \left(\alpha_{\mathrm{P}}^{(\mathrm{a+1})} \right)_{\mathrm{f}} - \left[\phi_{\mathrm{f}}^{(\mathrm{a})} - \phi_{\mathrm{mf}}^{(\mathrm{a})} \right] \mathscr{I} \left(\alpha_{\mathrm{P}}^{(\mathrm{a})} \right)_{\mathrm{f}} \right\| \leq M \Delta t$$

Then, the following relation is verified:

$$\forall a \in [|0, n-1|], \left\| \left[\phi_{\mathrm{f}}^{(\mathrm{a})} - \phi_{\mathrm{mf}}^{(\mathrm{a})} \right] \mathscr{I} \left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})} \right)_{\mathrm{f}} - F_{\alpha_{\mathrm{f}}}^{(\mathrm{a})} \right\| \leq (n-a) M \Delta t$$

Beside, considering a constant time step, the backward coefficients c_{n}, and $c_{\mathrm{n\text{-}2}}$ are positive constants. Defining the positive constant $\gamma = \dfrac{c_{\mathrm{n\text{-}2}}}{c_{\mathrm{n}}}$, according to Def. 3.136 and Def. 3.139:

$$F_{\mathrm{bck_f}}^{(\mathrm{n})} = \sum_{k=1}^{n} \gamma^k F_{\alpha_{\mathrm{f}}}^{(\mathrm{n\text{-}k})}$$

and

$$\phi_{\mathrm{bck_f}}^{(\mathrm{n})} = \sum_{k=1}^{n} \gamma^k \left(\phi_{\mathrm{f}}^{(\mathrm{n\text{-}k})} - \phi_{\mathrm{mf}}^{(\mathrm{n\text{-}k})} \right)$$

Consequently:

$$\left\| \phi_{\mathrm{bck_f}}^{(\mathrm{n})} \mathscr{I} \left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})} \right)_{\mathrm{f}} - F_{\alpha_{\mathrm{f}}}^{(\mathrm{n\text{-}1})} \right\| = \sum_{k=1}^{n} \gamma^k \left(\left[\phi_{\mathrm{f}}^{(\mathrm{n\text{-}k})} - \phi_{\mathrm{mf}}^{(\mathrm{n\text{-}k})} \right] \mathscr{I} \left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})} \right)_{\mathrm{f}} - F_{\alpha_{\mathrm{f}}}^{(\mathrm{n\text{-}k})} \right)$$

12. It means that the algorithm used for the resolution for previous time steps remains at least first-order.

and then:

$$\left\| \phi_{\mathrm{bck_f}}^{(\mathrm{n})} \mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}\right)_{\mathrm{f}} - F_{\alpha_{\mathrm{f}}}^{(\mathrm{n\text{-}1})} \right\| \leq \sum_{k=1}^{n} \gamma^k \left\| \left[\phi_{\mathrm{f}}^{(\mathrm{n\text{-}k})} - \phi_{\mathrm{mf}}^{(\mathrm{n\text{-}k})}\right] \mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}\right)_{\mathrm{f}} - F_{\alpha_{\mathrm{f}}}^{(\mathrm{n\text{-}k})} \right\|$$

$$\left\| \phi_{\mathrm{bck_f}}^{(\mathrm{n})} \mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}\right)_{\mathrm{f}} - F_{\alpha_{\mathrm{f}}}^{(\mathrm{n\text{-}1})} \right\| \leq M \Delta t \sum_{k=1}^{n} k \gamma^k$$

$$\left\| \phi_{\mathrm{bck_f}}^{(\mathrm{n})} \mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}\right)_{\mathrm{f}} - F_{\alpha_{\mathrm{f}}}^{(\mathrm{n\text{-}1})} \right\| \leq M \left[\frac{\gamma - \gamma^{n+1}}{(1-\gamma)^2} - \frac{n\gamma^{n+1}}{1-\gamma}\right] \Delta t$$

With a constant time step $\gamma = 1/3$ so:

$$\lim_{n \to +\infty} \left[\frac{\gamma - \gamma^{n+1}}{(1-\gamma)^2} - \frac{n\gamma^{n+1}}{1-\gamma}\right] = \frac{3}{4}$$

and:

$$\left\| \phi_{\mathrm{bck_f}}^{(\mathrm{n})} \mathscr{I}\left(\alpha_{\mathrm{P}}^{(\mathrm{n,m,0})}\right)_{\mathrm{f}} - F_{\alpha_{\mathrm{f}}}^{(\mathrm{n\text{-}1})} \right\| = O(\Delta t)$$

3.6 Reference numerical set-up

This section synthesizes the reference numerical parameters used in this document. They are based on the best OpenFOAM practices used in the literature. In the following of the document, each study takes this numerical set-up as a reference with few modifications depending on the objective.

3.6.1 Numerical schemes

a) Time schemes

Numerical parameter		Reference values

`ddtScheme`	`default`	`Euler`
		`CranckNicolson 1`
		`CranckNicolson 0.95`
		`backward`

b) Face interpolation schemes

Table 3.7 – Tested temporal time schemes along the numerical studies.

Numerical parameter		Reference values
interpolationSchemes	default	linear
divSchemes	default	Gauss linear
	div(rhoPhi,U)	Gauss linearUpwindV grad(U)
	div(phi,alpha)	Gauss vanLeer
	div(phir,alpha)	Gauss upwind
	div(rhoPhi,k)	Gauss linearUpwind limitedGrad
	div(rhoPhi,omega)	Gauss linearUpwind limitedGrad
laplacianSchemes	default	Gauss linear corrected

c) Cell-centered gradient schemes

Table 3.8 – Tested cell-centered gradient schemes along the numerical studies.

Numerical parameter		Reference values
gradSchemes	default	cellLimited leastSquares 1

d) Surface normal gradient schemes

Table 3.9 – Tested surface normal gradient schemes along the numerical studies.

Numerical parameter		Reference values
snGradSchemes	default	corrected
laplacianSchemes	default	Gauss linear corrected

3.6.2 Numerical algorithms

a) Matrix-system resolution

Table 3.10 – Tested matrix-system resolution parameters along the numerical studies.

Solved field	Parameter	Reference values
p_rgh	solver	GAMG
	smoother	DIC
	tolerance	1e-10
	relTol	0
U; k; omega	solver	PBiCGStab
	preconditioner	DILU
	tolerance	1e-10
	relTol	0
alpha	solver	smoothSolver
	smoother	symGaussSeidel
	tolerance	1e-10
	relTol	0
cellDisplacement	solver	GAMG
	preconditioner	GaussSeidel
	tolerance	1e-7
	relTol	0

b) PIMPLE and PISO algorithm

Table 3.11 – Tested PIMPLE and PISO parameters along the numerical studies.

Parameter	Reference values
momentumPredictor	yes
nOuterCorrectors	10
nCorrectors	3
nNonOrthogonalCorrectors	1
correctPhi	no

c) VOF resolution

Parameter	Reference values
nAlphaCorr	2
cAlpha	0.5
firstOrderPredictor	yes
nLimiterIter	50
clip	yes

PART II

foamStar numerical model accuracy

This part presents an analysis of some specific features of the numerical model previously described. It is a continuation of the work done by Kim (2021). In order to progressively reach naval simulations with waves, the successively studied cases are, 2D Taylor-Green vortices to check the behavior without air/water interface, the 2D wave regular wave propagation in periodic domain to check the behavior without relaxation zones, and finally 2D wave propagation with relaxation zones at the inlet and outlet. The objectives are to verify the selected models, to compare the time schemes and to establish some recommendations on the mesh structure. In order to maintain consistency, the resolution algorithm presented in Section 3.3 is used all along this part.

CHAPTER 4

TAYLOR-GREEN VORTICES

The objective this first study is to verify the order of convergence of a set of temporal and spatial schemes using a 2D mono-fluid problem and for various mesh characteristics. The 2D viscous Taylor-Green vortices is a standard unsteady mono-fluid case commonly used for the qualitative analysis of numerical schemes; this is because, with a given initial condition, the analytic solution of the incompressible Navier-Stokes equations is known for this problem, in which the field is unsteady and rotational.

4.1 Case definition

The simulated case is the same as the one used by Kim (2021). The 2D computational domain is in the (x,y) plane with coordinates $x, y \in [-\pi, \pi]$. The boundary conditions are periodic (`cyclic` type) at each side of the domain. The kinematic viscosity of the fluid is given as $\nu = 2\pi$ m^2.s^{-1} and the density as $\rho = 1$ kg.m^{-3}. Defining u_x and u_y as the components coordinates of the velocity $\mathbf{u}$, respectively along x and y axis, the initial conditions are:

$$u_\mathrm{x}^{(0)}(x,y) = -\sin(x)\cos(y) \tag{4.1a}$$

$$u_\mathrm{y}^{(0)}(x,y) = \cos(x)\sin(y) \tag{4.1b}$$

$$p^{(0)}(x,y) = \frac{\rho}{4}[\cos(2x)+\cos(2y)] \tag{4.1c}$$

With these initial conditions, the analytical solution at time t is:

$$u_\mathrm{x}^\mathrm{A}(x,y,t) = -\sin(x)\cos(y)\,e^{-2\nu t} \tag{4.2a}$$

$$u_\mathrm{y}^\mathrm{A}(x,y,t) = \cos(x)\sin(y)\,e^{-2\nu t} \tag{4.2b}$$

$$p^\mathrm{A}(x,y,t) = \frac{\rho}{4}[\cos(2x)+\cos(2y)]\,e^{-2\nu t} \tag{4.2c}$$

In the following the reference quantities are $L = 2\pi$ m, $U_{\text{ref}} = 1$ m.s^{-1} and $p_{\text{ref}} = 1$ Pa.

a) Error definition

For this study, the error $\mathscr{E}(t)$ is defined using the L_2 norm of the weighted errors at cell-centers positions $(\mathbf{x}_i)_{i=1}^{N_{\text{cells}}}$ compared to the analytical solution ψ^{a}. To take into account the non-uniformity of the mesh, the errors are weighted by the volume of the cells. The total error is normalized using the analytical maximal value over the domain. The mathematical formulation of the error is:

$$\mathscr{E}(t) = \frac{\sqrt{\sum_{i=1}^{N_{\text{cells}}} \left[\psi(\mathbf{x}_i,t) - \psi^{\text{A}}(\mathbf{x}_i,t)\right]^2 V_i^2}}{\max\limits_{1\leq i\leq N_{\text{cells}}} |\psi^{\text{A}}(\mathbf{x}_i,t)|\ V_\Omega} \tag{4.3}$$

with V_Ω the volume of the numerical domain [1]:

$$V_\Omega = \sum_{i=1}^{N_{\text{cells}}} V_i \tag{4.4}$$

In the following, a particular notation $\mathscr{E}_f(t)$ is used in order to identify the error obtained with the finest refinement. The related refinement is to be specified case by case when the notation is used. This *finest* error is used to calculate an intrinsic error $\mathscr{E}(t) - \mathscr{E}_f(t)$. Assuming that the numerical error converges towards a non-zero value $\mathscr{E}_\infty(t) \neq 0$, the intrinsic error is used to remove this non-null limit form the error $\mathscr{E}$. This "intrinsic" error will especially be useful to study separately the time and space convergence, making disappear the potential saturation error due to the fixed space resolution when looking at the time convergence or, respectively, due to the fixed time resolution when looking at the space convergence. Note that in this context where the analytical solution is known and its frequency content simple, it is not necessary to use an advanced V$V procedure to study the convergence.

4.2 Influence of the time schemes on the convergence

The objective is to compare the accuracy of three time schemes available within OpenFOAM-5 implicit Euler (`EULER`), Crank-Nicolson (`CN1`), and second-order backward [2]

1. Even if the simulation is done with a 2D assumption, with OpenFOAM implementation, the mesh has to be a 3D mesh, so an arbitrary thickness is given along the z direction. The weighting process using the resulting volume is equivalent to a surface weighting in the (x,y) plane.

2. The OpenFOAM-5 backward scheme was slightly modified due to a non-accurate scheme formulation for the first time-step in the context of this study.

(`BCK`). In the following the term backward is used to name the second-order backward. As mentioned in Section 3.2, the theoretical order of convergence for these schemes are one for Euler and two for Crank-Nicolson and backward. To control the convergence of the scheme a regular mesh with 256 by 256 squared cells of size $\Delta x = \dfrac{2\pi}{256}$ is used. For this study the mesh is labeled `U256x256`. Table 4.1 synthesizes the set of time steps (from 0.004 s to 0.4 s) with the associated Courant Number $\mathrm{Cfl}_x = U_{\mathrm{ref}} \dfrac{\Delta t}{\Delta x}$. This test matrix is directly extracted from the work of Kim (2021).

Table 4.1 – Test matrix for Taylor-Green time-convergence study with `U256x256` mesh.

Time step Δt [s]	**Courant Number Cfl_x**
0.004	0.16
0.01	0.41
0.02	0.81
0.04	1.63
0.08	3.24
0.1	4.07
0.125	5.09
0.2	8.15
0.25	10.2
0.4	16.3

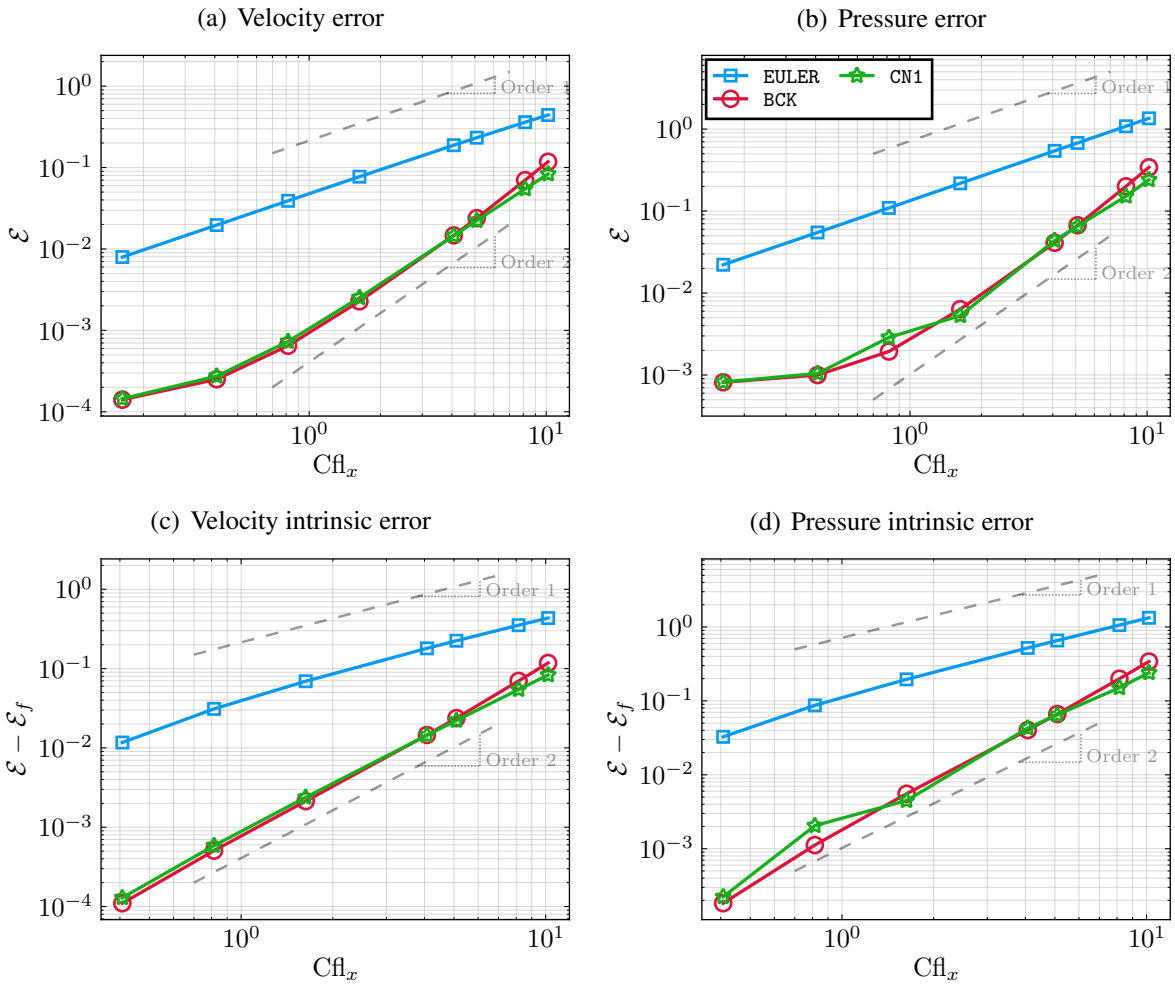


Figure 4.1 – Temporal convergence of the velocity and pressure error for various temporal schemes at $t = 1$ s with `U256x256`.

The velocity and pressure errors at $t = 1$ s are plotted with respect to the $\mathrm{Cfl_x}$ in Figure 4.1. The logarithmic scale is used for both the vertical and horizontal axis. The finest resolution used for the evaluation of $\mathscr{E}_f$ is $\Delta t = 0.004$ s. Each data series corresponds to one time scheme and the dotted lines show the expected slopes for a first and a second-order convergence. The results obtained using `Euler` and `CN1` are consistent with Kim (2021). The saturation phenomenon occurring for $\mathrm{Cfl_x} < 0.5$ in the top plots is due to the prevalence of the spatial-discretization error. In the zone where the temporal-discretization error is dominant the slopes of the straight part are consistent with the theoretical order of the studied numerical schemes. The slope of the series obtained with Euler scheme fits with the first order slope. For `BCK` and `CN1` series, the trend in the zone $\mathrm{Cfl_x} > 0.5$ is close to the expected second-order slope

but some irregularities can be noticed. For the `CN1` series, the irregularities close to $\mathrm{Cfl_x} = 1$ on the pressure-convergence plot were noted by Kim (2021) already. The intrinsic error is used to remove the spatial-discretization error assumed to be constant regardless of the temporal discretization. The intrinsic error confirms the accordance between the theoretical order of convergence and the numerical results. The errors obtained with `CN1` and `BCK` schemes are close to each other.

4.3 Influence of mesh deformations on the spatial convergence with and without skewness corrector

The next step is to verify the accuracy of a given numerical set of spatial schemes taking into account the mesh quality. In this section, the `skewCorrector` option is tested to see potential improvements. As developed in Section 3.1.3, the skewness correction theoretically reduces the loss of accuracy for face center interpolations. Moreover, some past attempts to run naval simulation with foamStar indicate that the skewness is a tricky issue especially with viscous boundary layers.

4.3.1 Mesh types

Four mesh types are studied: `regular`, `stretch`, `twist` and `zigZag`. The `regular` meshes are built with a uniform grid of Cartesian square cells as the one used in the previous section. The `stretch` mesh is built with a uniform grid of Cartesian cells with an aspect ratio $\mathrm{AR} = \dfrac{\Delta x}{\Delta z}$ above one. The `twist` and `zigZag` meshes consist in applying a specific transformation to the nodes coordinates of a regular mesh. In order to assess the quality of the meshes, three characteristics are controlled: the aspect ratio, the non-orthogonality (Sec. 3.1.6) and the skewness (Sec. 3.1.3).

a) Regular mesh

The `regular` meshes are used as references in order to control the accuracy and the convergence order of the spatial discretization. In order to remain consistent with the time integration, it was decided to run the spatial convergence at fixed $\mathrm{Cfl_x}$ and only with the Crank-Nicolson scheme. For a deformed mesh the reference $\mathrm{Cfl_x}$ is the one calculated with the non-deformed regular mesh. Table 4.2 gives the list of the spacial discretization studied and

the associated time discretization. The horizontal and vertical resolutions N_x and N_y are related to the cell size by the relations

$$N_x = \frac{2\pi}{\Delta x} \tag{4.5a}$$

$$N_y = \frac{2\pi}{\Delta y} \tag{4.5b}$$

Table 4.2 – `Regular` meshes used on the Taylor-Green vortices case.

Name	**Horizontal discretization** N_x	**Vertical discretization** N_y	Δt [s]
`U8X8`	8	8	0.016
`U16X16`	16	16	0.008
`U32X32`	32	32	0.004
`U64X64`	64	64	0.002
`U128X128`	128	128	0.001
`U256X256`	256	256	0.0005

b) Stretched mesh

The `stretch` type is used in order to control the influence of the cells aspect ratio on the accuracy of a simulation without interface. Table 4.3 lists the different meshes used in this study. The aspect ratio is kept above one, it means that the smaller dimension between Δx and Δy is always the vertical one Δy. A representation of the mesh `U16X64` is shown in Figure 4.2.

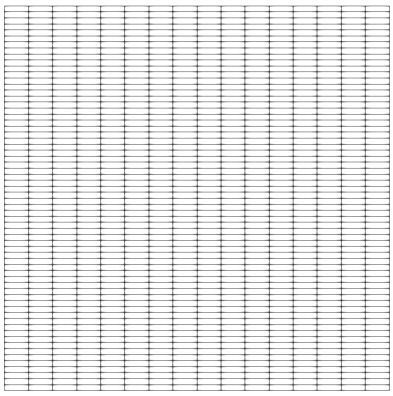


Figure 4.2 – `stretch` type mesh `U16X64`.

Table 4.3 – `Stretch` meshes used on the Taylor-Green vortices case.

Name	N_x	N_y	**AR**	Δt [s]
`U8X16`	8	8	2	0.016
`U16X132`	16	32	2	0.008
`U32X64`	32	64	2	0.004
`U64X128`	64	128	2	0.002
`U128X256`	128	256	2	0.001
`U8X32`	8	32	4	0.016
`U16X64`	16	64	4	0.008
`U32X128`	32	128	4	0.004
`U64X256`	64	256	4	0.002
`U8X64`	8	64	8	0.016
`U16X128`	16	128	8	0.008
`U32X256`	32	256	8	0.004
`U8X128`	8	128	16	0.016
`U16X256`	16	256	16	0.008
`U8X256`	8	256	32	0.016

c) Twisted mesh

The `twist` deformation aims to reproduce a strong mesh deformation that could be induced by a pitching body with a morphing mesh. In the deformation equations, (x_0, y_0) are the coordinates of a mesh node before the deformation when (x, y) are the coordinates of the same mesh node after the deformation.

For simplicity and because of the absence of body, the mesh morphing algorithm is not used. The deformation consists in rotating the mesh nodes around the center of the domain with a maximal rotation angle $\theta_{\max}$ at the center $(x_0, y_0) = (0,0)$ and an angle $\theta = 0°$ at the boundaries of the domain $(x_0, y_0) \in \{-\pi, \pi\}^2$. Along both vertical and horizontal axis the rotation angle follows a sinusoidal law relatively to the position (x_0, y_0) in the non-deformed mesh. The coordinates (x, y) of deformed-mesh nodes follow the equations:

$$\theta(x_0, y_0) = \frac{\theta_{\max}}{4}\left[1+\cos\left(\frac{2\pi x_0}{L}\right)\right]\left[1+\cos\left(\frac{2\pi y_0}{L}\right)\right] \tag{4.6a}$$

$$x(x_0, y_0) = x_0\cos\left(\theta(x_0, y_0)\right) - y_0\sin\left(\theta(x_0, y_0)\right) \tag{4.6b}$$

$$y(x_0, y_0) = -x_0\sin\left(\theta(x_0, y_0)\right) + y_0\cos\left(\theta(x_0, y_0)\right) \tag{4.6c}$$

For each uniform regular mesh from Tab. 4.2, the tested deformations are $\theta_{\max} \in$

$\left\{ \frac{k\pi}{8} \text{ with } k = 1..8 \right\}$. A representation of the mesh `U64X64` twisted with $\theta_{\max} = \pi$ is shown in Figure 4.3. The maximal value $\theta_{\max} = \pi$ is voluntary extreme as it would correspond to a ship flipped upside-down. However, it is interesting to study the convergence of the scheme with highly deformed cells.

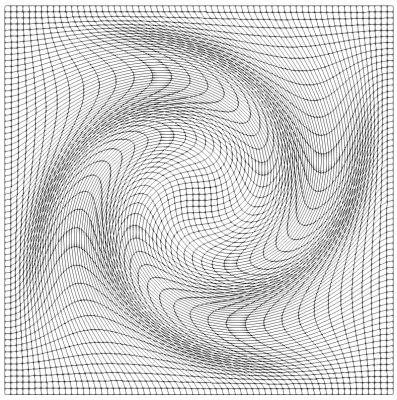


Figure 4.3 – `twist` deformation $\theta_{\max} = \pi$ applied on a `U64X64` mesh.

As mentioned, in this study, the quality of a twisted mesh is measured calculating the average and maximal values of three quantities: aspect ratio, non-orthogonality and skewness. For a spatial convergence study, it is expected that for a given level of deformation, the three quantities remain as similar as possible regardless of the spatial resolution. Figure 4.4 shows the aspect ratio evolution for different twist angles and mesh resolutions. The maximal aspect ratio increases significantly with the mesh refinement and the twist angle, reaching a value of 11.1 for the mesh `U256X256` twisted with $\Theta_{\max} = \pi$. The average aspect ratio increases with the twist angle but remains quite stable with resolutions above $N_x = 16$, reaching a value of 2.3 for the mesh `U256X256` twisted with $\Theta_{\max} = \pi$.

Figure 4.5 shows the non-orthogonality evolution for different twist angles and mesh resolutions. The maximal and average values increase with the twist angle. With OpenFOAM it is commonly admitted that cells with a non-orthogonality above 70° are highly non-orthogonal and of bad quality (OpenFOAM, 2022c). Within the range of considered twist angles, the maximal local non-orthogonality reaches 81° and the maximal average is 49° for the largest twist angle $\Theta_{\max}$. However, the maximal local non-orthogonality and the average non orthogonality are reasonably constant regardless of the spatial resolution so it means that the `twist` deformation is adequate to investigate the effect of non orthogonality.

Finally, Figure 4.6 shows the skewness evolution. Both the maximal and average skewness increase when the twist angle increases and decrease when the refinement increases.

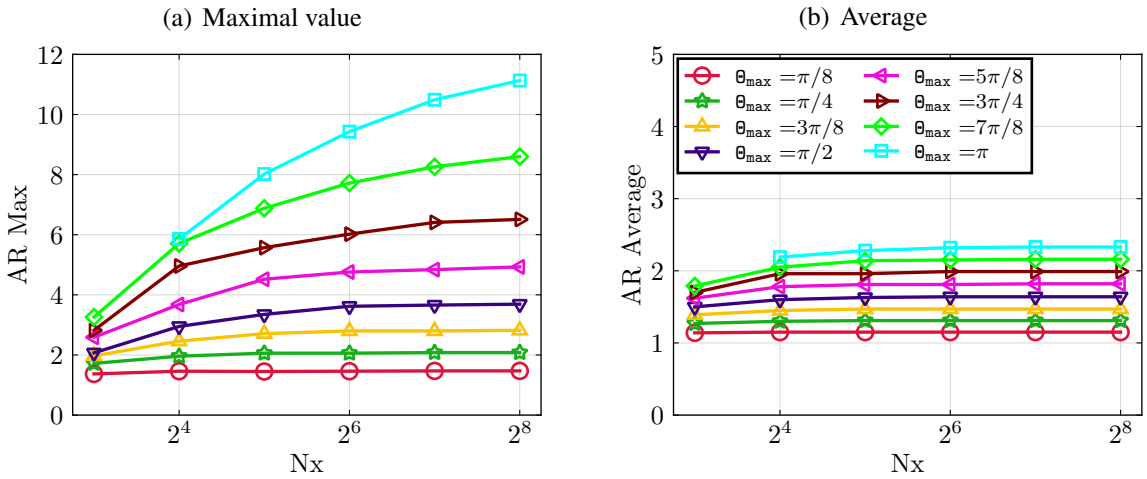


Figure 4.4 – Evolution of the maximal and average aspect ratios depending on the mesh resolution for various maximal angles of `twist` deformation.

With OpenFOAM, a skewness above 4 is assumed to badly impair the simulation (OpenFOAM, 2022c). This limit is never reached and the averaged skewness reaches values below 0.1 for mesh resolutions above $N_x = 32$. Consequently, the selected `twist` deformation is assumed to keep the skewness small.

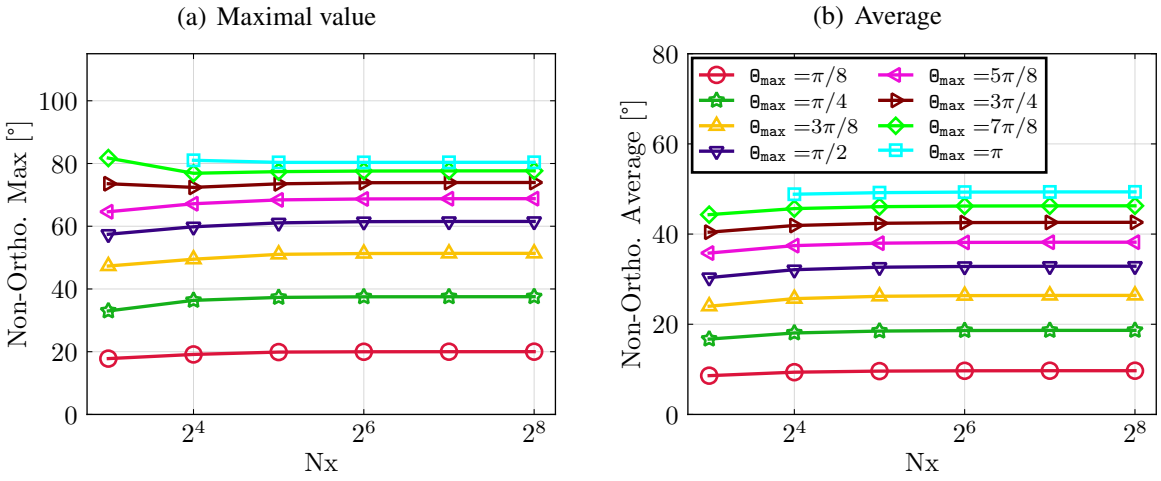


Figure 4.5 – Evolution of the maximal and average non-orthogonality angle depending on the mesh resolution for various maximal angles of `twist` deformation.

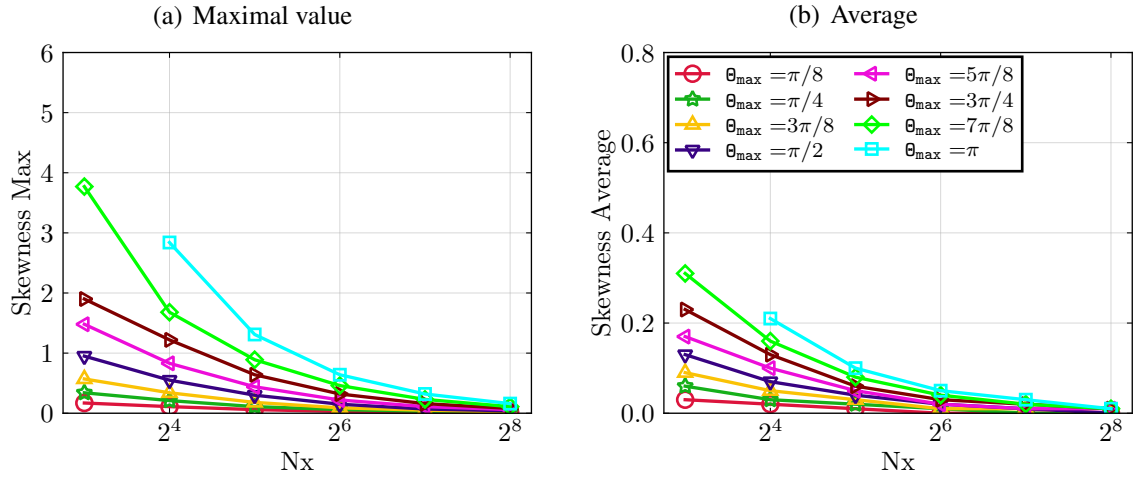


Figure 4.6 – Evolution of the maximal and average mesh skewness depending on the mesh resolution for various maximal angles of `twist` deformation.

d) Zig-zag mesh

The `zigZag` deformation is used to highlight the influence of the skewness on the spatial schemes accuracy. It consists in moving alternatively up and down the mesh nodes along the vertical axis. The largest amplitude of this oscillation is noted $A_{\max}$ and is applied along the median line $y = 0$ m. No displacement is applied to the boundary nodes. The oscillation amplitude follows a sinusoidal law along the vertical axis and remains constant along the horizontal one. The coordinates (x, y) of the deformed-mesh nodes follow the equations:

$$A_{\max} = \eta \Delta x \tag{4.7a}$$

$$A(x_0, y_0) = \frac{A_{\max}}{2}\left[1 + \cos\left(\frac{2\pi y_0}{L}\right)\right] \tag{4.7b}$$

$$x(x_0, y_0) = x_0 \tag{4.7c}$$

$$y(x_0, y_0) = A(x_0, y_0)\cos\left(\frac{\pi y_0}{\Delta x}\right) \tag{4.7d}$$

In order to maintain the consistency between the mesh resolutions, the amount of deformation is characterized by the non-dimensional coefficient η equal to the ratio between the maximal point displacement and the cell size Δx as follows:

$$\eta = \frac{\max\limits_{1 \leq i \leq N_{\text{nodes}}} \left| y_i - y_{0,i} \right|}{\Delta x} \tag{4.8}$$

For each uniform regular mesh from Tab. 4.2, the tested deformations are $\eta \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.75, 1\}$. A representation of the mesh `U32X32` deformed with $\eta = 0.75$ is shown in Figure 4.7.

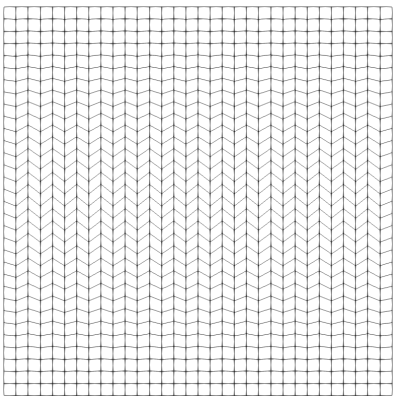


Figure 4.7 – `zigZag` deformation applied on a `U32X32` mesh.

As for the `twist` deformation, the impact of `zigZag` deformation on the mesh quality is measured using the aspect ratio, the non-orthogonality and the skewness. Figure 4.8 shows the evolution of the maximal and average aspect ratios depending on the mesh resolution for several amplitudes of `zigZag` deformations. For a given amplitude both the maximal and average values are rather independent of the mesh resolution. The aspect ratios increase with the amplitude of the deformation but the maximal local value remains below 5 and the maximal average below 3.

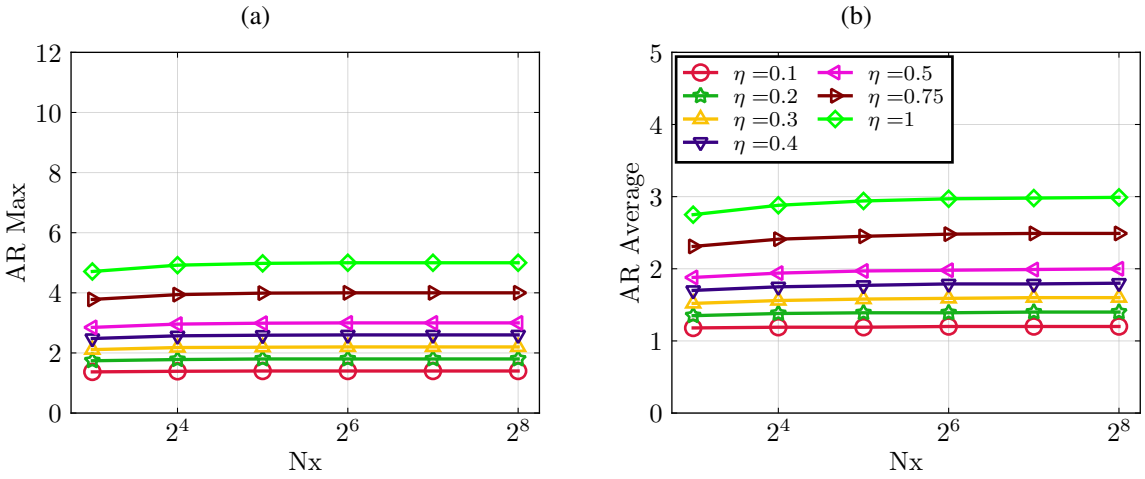


Figure 4.8 – Evolution of the maximal and average aspect ratio depending on the mesh resolution for various coefficients of `zigZag` deformation.

Figure 4.9 shows the impact of the `zigZag` deformation over the non-orthogonality. Within the selected range of `zigZag` amplitudes, the evolution of the maximal non-orthogonality is similar to the one observed with the `twist` deformation (Fig. 4.5). The average value respects the expected independence of the mesh resolution and remains below 40°.

Finally, Figure 4.10 shows the impact of the `zigZag` deformation on the skewness. Compared to the mesh analysis done with the `twist` deformation (Fig. 4.6), the `zigZag` deformation meets the objective of getting a sufficiently constant evolution for resolutions above $N_x = 32$. However, because of the progressive deformation from the boundaries to the center of the mesh, the average skewness remains below 0.5 with resolutions above $N_x = 32$. For the finest mesh ($N_x = 256$), the maximal local skewness reaches the targeted value of 4.

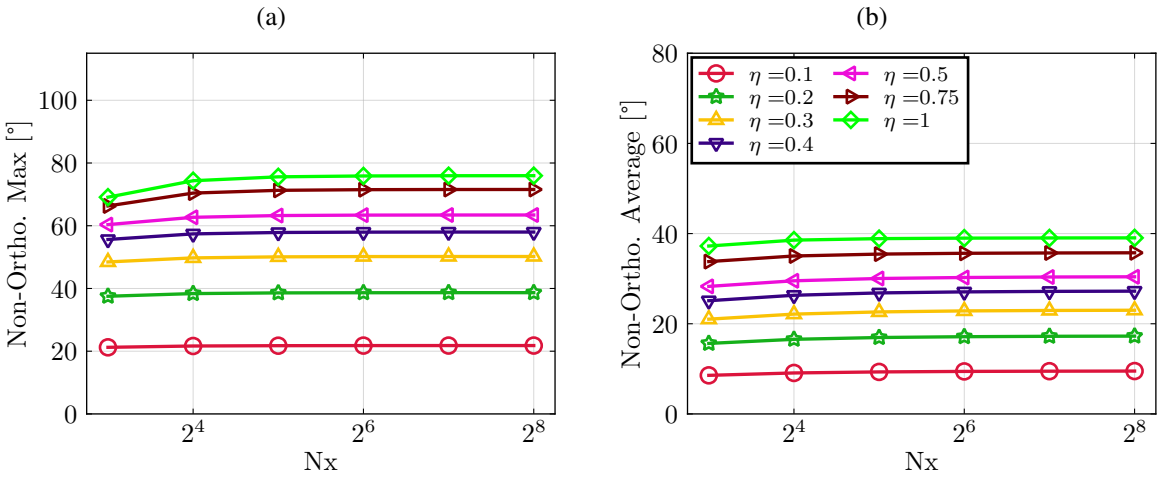


Figure 4.9 – Evolution of the maximal and average non-orthogonality angle depending on the mesh resolution for various coefficients of `zigZag` deformation.

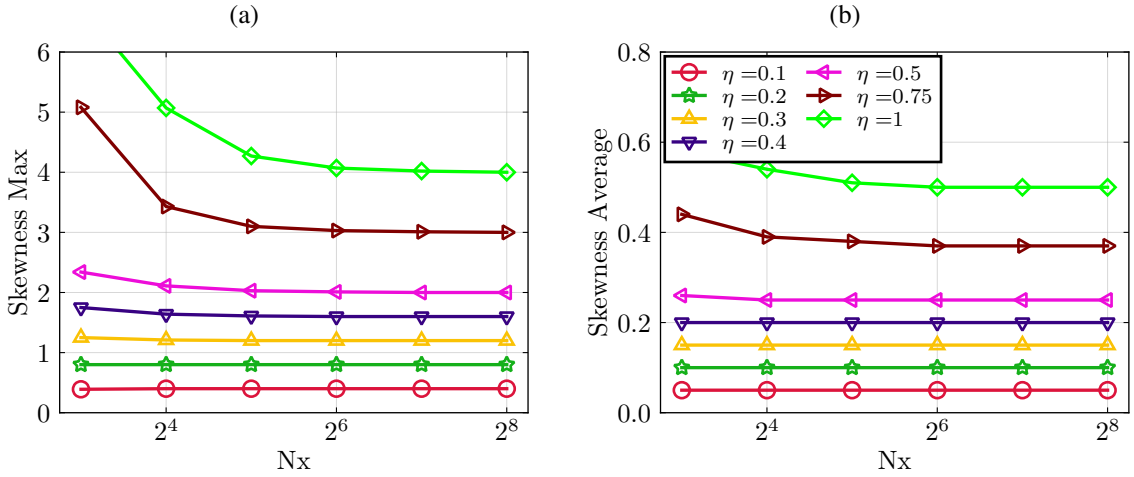


Figure 4.10 – Evolution of the maximal and average mesh shewness depending on the mesh resolution for various coefficients of `zigZag` deformation.

e) Summary

Three mesh deformations have been designed to affect specific characteristics of the mesh quality. Table 4.4 synthesizes the influence of each deformation. In the following the `stretch` deformation is used to highlight the impact of the aspect ratio on the mesh convergence. The `twist` deformation is convenient to highlight the influence of non-orthogonality. Finally, only the `zigZag` deformation has a strong influence on the skewness.

Table 4.4 – Influence of the mesh deformations on the mesh quality

Deformation	Aspect ratio	Non-orthogonality	Skewness
`stretch`	High	0	0
`twist`	Small	High	Small
`zigZag`	Small	High	High

4.3.2 Results

a) Stretched mesh (aspect ratio influence)

The set of `stretch` meshes is considered first to isolate the effect of the aspect ratio. Figure 4.11 shows the effect of stretching the mesh on the spatial convergence of the error $\mathscr{E}$ on the velocity plot (a) and the pressure field plot (b). For a given longitudinal cell size, the velocity error remains similar regardless of the cell aspect ratio. For the pressure (b), high aspect ratio tends to increase the error. For aspect ratios AR ≤ 8, at least three vertical refinements are tested. The slopes are consistent with the expected second-order. Finally, this study shows that for the simulation of Taylor-Green vortices, with the selected numerical setup, *foamStar* remains of second-order in space, at least when AR ≤ 8 and the impact of the stretched mesh on the numerical error is weak.

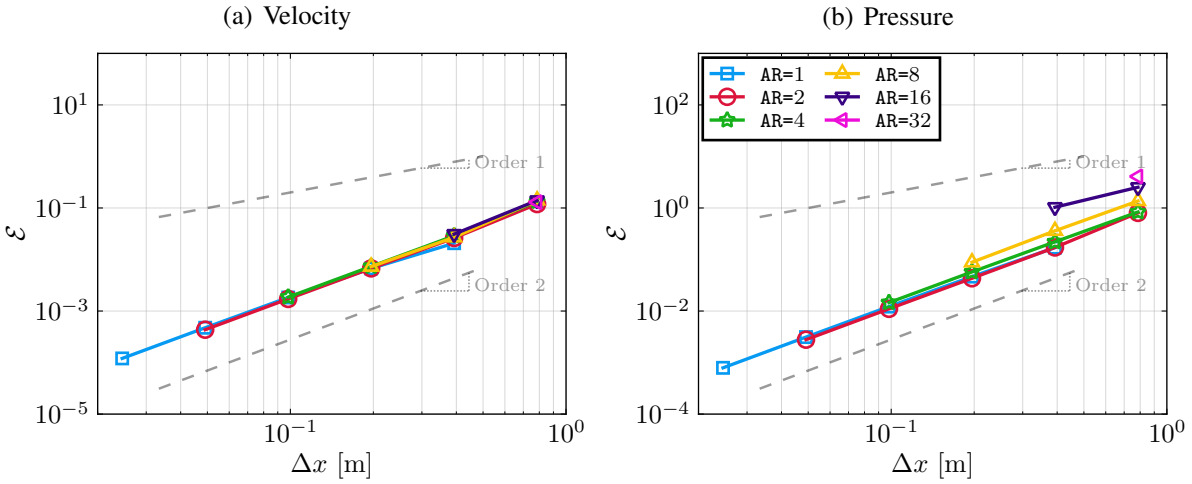


Figure 4.11 – Spatial convergence of the velocity and pressure error for various aspect ratios *AR* using the `stretch` deformation.

b) Twisted mesh (non-orthogonality influence)

As seen previously the `twist` deformation generates locally some high aspect ratio and a significant maximal and average non-orthogonality in the meshes. Thanks to the result of the previous paragraph, the impact of the aspect ratio can be neglected and the prevalent mesh characteristics impacting the accuracy is most probably the non-orthogonality (with eventually also the local skewness for coarse meshes). Figure 4.12 shows the velocity and pressure errors for several `twist` angles and mesh resolutions. The error increases significantly with the `twist` angle. Nevertheless, whatever the deformation angle is, the slope of the curve is close to the expected second-order slope. Some spatial refinements are not plotted for high `twist` angles ($\Theta \geq \frac{3\pi}{8}$) because those simulations stopped with computational error before the targeted $t = 1$ s. As shown, both the aspect ratio and the skewness decrease when the mesh resolutions increases whereas the non-orthogonality remains constant. Then, the failed simulations are probably caused by other mesh characteristics badly impaired by the `twist` deformation which have not been identified in this study. However, it can be assessed that with the current numerical setup, the non-orthogonality caused by the tested ranges of `twist` deformations and the small skewness do not affect the spatial order of convergence. Note that the current numerical setup includes a non-orthogonal correction. The effect of this non-orthogonal corrector is not studied in this section and it is kept as the default parameter.

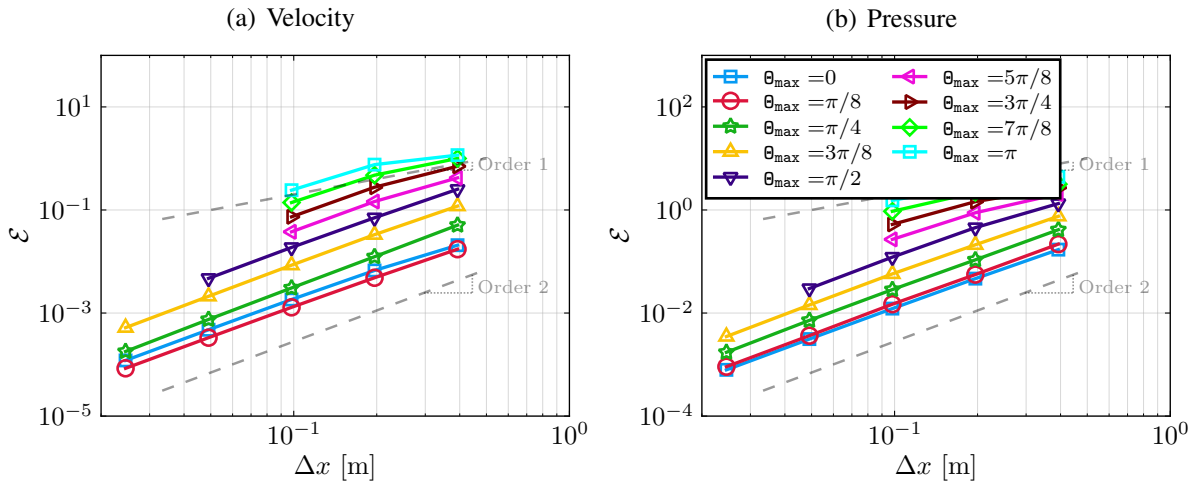


Figure 4.12 – Spatial convergence of the velocity and pressure error for various maximal angles of `twist` deformation.

c) Zig-Zag mesh (Skewness impact with and without skewness corrector)

The third deformation studied is the `zigZag`, dedicated to the analysis of the impact of the cell skewness. Figure 4.13 shows the velocity and pressure error for several mesh resolutions and `zigZag` coefficients. The OpenFOAM skewness corrector is also tested. Fig. 4.12(a) and Fig. 4.12(b) show the results obtained without skewness correctors when Fig. 4.12(c) and Fig. 4.12(d) show the results with the skewness corrector.

Without skewness corrector, the numerical values obtained with this type of deformation do not converge towards the targeted analytical field. Even for the smallest `zigZag` coefficient $\eta = 0.1$, the deformations impair significantly the simulations. With the finest resolutions, the velocity error is more than 100 times larger with $\eta = 0.1$ than with the non-deformed mesh. About the pressure, for any tested deformation and mesh resolution, the error remains larger than 100%. Moreover, without the skewness corrector, the largest deformations cause the simulations failure before reaching the targeted $t = 1$ s.

Using a skewness corrector improves significantly the results. With this corrector the errors obtained with $\eta = 0.1$ are close to the ones obtained without deformations and the pressure error for the finest meshes remains below 10% up to $\eta = 0.3$. However, the saturation of the error is still noticeable.

It have been shown that the `zigZag` deformation induces some non-orthogonality. Comparing to the results obtained with `twist` deformation, the error obtained with the *zigZag* deformation is significantly higher than the error obtained with `twist` deformation. For

instance, the non-orthogonality induced by a `zigZag` coeficient $\eta = 0.5$ is of the same order as the non-orthogonality obtained with a `twist` angle $\theta_{\max} = \pi/2$ but the velocity error using the first mesh is about 10 times larger for the coarser meshes and more than 100 times larger with the resolution $N_x = 128$. So it is assumed that with `zigZag` deformation the error induced by the skewness prevails on the one induced by the non-orthogonality.

Figure 4.14 shows the corresponding velocity and pressure intrinsic errors ($\mathscr{E} - \mathscr{E}_f$). This removes the saturated error and it eases the qualitative observation of the order of convergence of the spatial discretization with the `zigZag` deformation. Given a `zigZag` coefficient, $\mathscr{E}_f$ represents the error obtained at $t = 1$ s with respect to the result using for the finest longitudinal refinement. With this intrinsic error it appears that the slopes remain consistent with the second-order accuracy. However, as mentioned, even with small deformations, the errors $\mathscr{E}$ are large, so it seems questionable to establish a second-order accuracy using this intrinsic error. It just implies that the scheme behaves as expected for the rest of the numerical errors but the global error is largely dominated byt the mesh skewness-induced error. It has to be kept in mind that the chosen deformations are rather extreme both in amplitude and spatial frequency.

Finally, with the `zigZag` deformation, the main conclusion is that the error is significant even with small deformations. With *foamStar*, the zigZag-like cells thus impair a lot the simulation compared to twisted cells or stretched ones. The second point to highlight is the efficiency of the skewness corrector when the `zigZag` distortion is moderate.

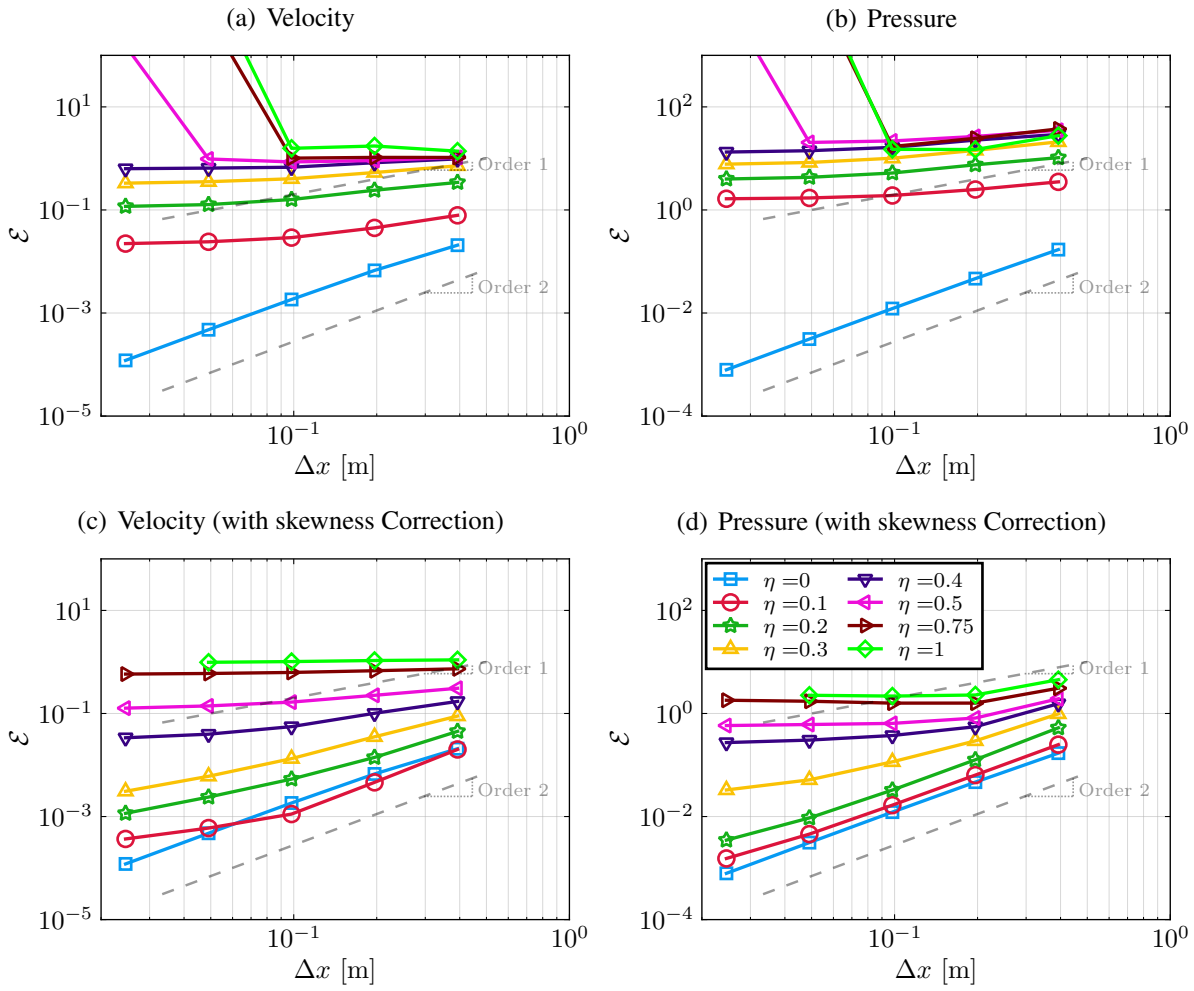


Figure 4.13 – Spatial convergence of the velocity and pressure error with and without OpenFOAM skewness corrector for various coefficients of `zigZag`.

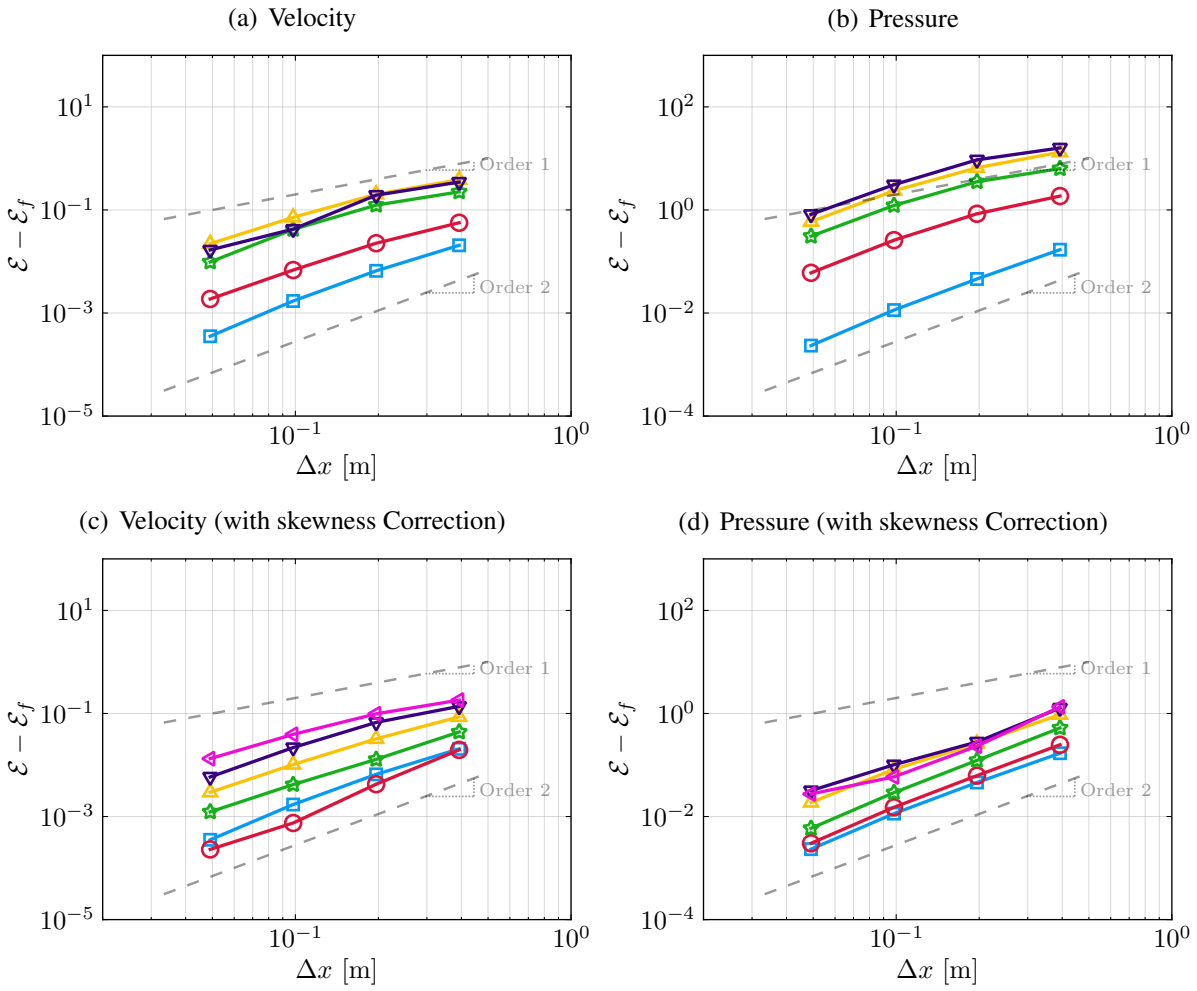


Figure 4.14 – Spatial convergence of the velocity and pressure intrinsic error with and without OpenFOAM skewness corrector for various coefficients of `zigZag`.

4.4 Conclusion

In this chapter, the accuracy of three time schemes has been studied. The order of convergence of the time schemes has been checked and then the effect of three mesh deformations on the spatial convergence has been investigated. It appears that for a Navier-Stokes simulation of Taylor-green vortices, both the temporal and spatial convergences are in accordance with the theoretical order of convergence of the tested schemes. It also results that, as expected, the Euler scheme is significantly less accurate than the Crank-Nicolson and second-order backward. Those two last second-order schemes have very similar accuracy.

Concerning the mesh deformations, `stretch` deformation with aspect ratios below 8 or 16 and reasonable `twist` deformation ($\Theta \leq \frac{3\pi}{8}$) do not impair significantly the simulations whereas even a small `zigZag` deformation drops down largely the accuracy. The skewness corrector limits a lot the influence of such a `zigZag` deformation when it remains of small amplitude.

CHAPTER 5

PERIODIC REGULAR WAVE

The previous analysis was dedicated to investigate the behavior of the numerical schemes in a periodic monofluid domain. The next step towards a naval simulation is to consider a set of multi-phase flow problems. The case exposed in this chapter is a regular wave propagating in a one wavelength long periodic bi-dimensional domain. This study completes the work done by Choi et al. (2020) and Kim et al. (2022a) based on the same problem. As mentioned in Section 3.3.2, the `MULES` algorithm is used for the VOF resolution. The objective is to compare the accuracy of the Crank-Nicolson `MULES` model with the newly implemented backward `MULES` model (Sec. 3.5.1) for 2D pure wave propagation. In order to progressively reach a mesh design commonly used for naval applications, some non-uniform meshes are tested with progressive cell refinements from the bottom and top boundaries towards the free surface.

The characteristics of the simulated waves are synthesized in Table 5.1. In the following the labels `H01`, `H005` and `H001` refer to the selected waves with the wave height $H = 0.1$ m, $H = 0.05$ m and $H = 0.01$ m, respectively. Within the fully nonlinear wave framework, imposing the depth D, the wave height H and the wavelength λ characterizes completely the wave. The wave periods in Table 5.1 are obtained using the stream function wave theory (Rienecker and Fenton, 1981) formulation implemented in the in-house tool *CN-Stream* (Ducrozet et al., 2019). For a given simulation, the selected wave field is imposed as an initial condition at $t = 0$ s. This initial field is based on the potential stream-theory solution and is only is imposed in the water phase. In the air ($\alpha = 0$), the velocity and pressure are null at $t = 0$ s. The 2D-domain configuration is illustrated by Figure 5.1. For all meshes, the mean free-surface level is located at $z = 0$ m. The simulations are conducted in the fixed-frame referential from $t = 0$ s to $t = 40T$, T being the considered wave period. No turbulent model is applied. In the following, in order to evaluate a Courant number $\mathrm{Cfl_x}$, the reference velocity for a given wave is

$$\mathbf{u}_{\mathrm{ref}} = \frac{\pi H}{T}. \tag{5.1}$$

The boundary conditions are synthesized in Table 5.2. For more details on these BCs see Section 3.3.6.

Table 5.1 – Regular wave characteristics.

Item	**Unit**	H001	H005	H01
Depth (D)	[m]	1	1	1
Wavelength (λ)	[m]	1	1	1
Wave period (T)	[s]	0.80	0.79	0.76
Wave height (H)	[m]	0.01	0.05	0.1
Wave steepness ($\varepsilon = H/\lambda$)		1 %	5 %	10 %

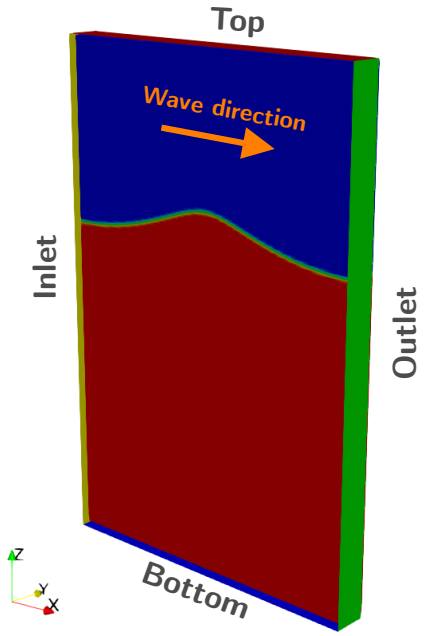


Figure 5.1 – 2D Periodic domain configuration for regular wave propagation.

Table 5.2 – Boundary conditions used for wave propagation in periodic domain.

Field	**Inlet\Outlet**	**Bottom**	**Top**
U	cyclic	slip	pressureInletOutletVelocity
p_rgh	cyclic	fixedFluxPressure	totalPressure
alpha	cyclic	zeroGradient	inletOutlet

5.1 Error definition

The accuracy of the numerical simulations is evaluated analyzing the damping in time of the wave elevation. The wave elevation is obtained by probes evenly spaced all along the simulation domain in x-direction. For a given probe located at the position x, and for the i^{th} time step, OpenFOAM returns the interpolated vertical position z verifying $\alpha^{(\mathrm{i})}(x,z) = 0.5$. This point is assumed to be the surface elevation $\eta(x,t^{(\mathrm{i})})$. For a given time t, a discrete Fourier transformation is applied on the sampled surface elevation at $x = 0$ m between $t_0 = t - 2.5T$ and $t_1 = t + 2.5T$. The resulting first harmonic amplitude is noted $A_1^{CFD}(t)$. This first harmonic amplitude of the surface elevation is compared to the analytic first harmonic amplitude obtained with *CN-Stream* and noted [1] A_1^{SF}. Then, the numerical error used in this section is defined as

$$\mathscr{E}(t) = \frac{A_1^{CFD}(t) - A_1^{SF}(t)}{A_1^{SF}(t)}. \tag{5.2}$$

As in Sec. a), the intrinsic errors $\mathscr{E}(t) - \mathscr{E}_f(t)$ are occasionally used.

5.2 Numerical set-up

The reference numerical set-up (Sec. 3.6) is used. The investigation regards some parameters relative to the VOF resolution and the time step, which are synthesized in Table 5.3. The numerical parameter `ddtScheme {default}` corresponds to the time scheme used for all the equations. Concerning the `MULES` type (`MULESType`), the `MULES` option refers to the full space and time `MULES` algorithm whereas the `S-MULES` option refers to the `MULES` algorithm applied on spatial schemes (see Sec. 3.3.2 for more details). Crank-Nicolson schemes are tested with two different off-center coefficients, 1 (`CN1`) and 0.95 (`CN95`). A range of time step (`deltaT`) from $t = T/25$ to $t = T/1600$ is tested.

1. SF exponent denotes the Stream-Function term

Table 5.3 – Numerical parameters studied with regular wave propagation in periodic domain.

Numerical parameter	Tested values

`ddtScheme {default}`	`CranckNicolson 1`
	`CranckNicolson 0.95`
	`backward`

`MULESType`	`MULES` `S-MULES`
`deltaT`	$\left\{ \frac{T}{25 \times 2^k} \mid k \in [\|1, 6\|] \right\}$

5.3 Meshes

Two types of meshes are designed:

Uniform regular mesh (`U`) A mesh of this type is perfectly uniform with isometric cells. The cell dimensions Δx and Δz are equal and constant over the mesh. As mentioned, the domain is one λ long in x-direction. In z-direction, the bottom boundary is located one λ below the mean free-surface level and the top boundary one λ above. The left plot of Figure 5.2 displays an example of uniform mesh with $\Delta x = 64$ (`U64x64`).

Non-uniform mesh (`NU`) A mesh of this type is divided into seven refinement zones[2]. The characteristics of the meshes are controlled by the cell dimensions Δx and Δz in the free surface zone (`ZONE 0`) and by the specific wave height `H`. In z-direction, the bottom boundary is located one λ below the mean free-surface level but the top boundary is $\lambda/2$ above. This limitation of the mesh high is motivated by the work presented by Kim (2021) and Kim et al. (2022a). The cells inside the free-surface zone are not necessarily isometric whereas all the cells outside the free-surface zone are isometric. In the free-surface zone, if the cells are non-isometric, the mesh is named *non-uniform stretched* mesh. The quantity $\mathrm{AR_{FS}} = \Delta x/\Delta z$ is the cell aspect ratio in the free-surface zone. Figure 5.3 is an example of non-uniform stretched mesh on which the refinement zones are identified. The geometrical characteristics of these zones are given in Table 5.4. When $\Delta z = \Delta x$ in the free-surface zone **0**, the zones **0**, **1** and **4** are mixed up and the mesh is named *non-uniform regular mesh*, see, e.g. right plot of Fig. 5.2.

In the following, the different types of meshes are tested with one or several of the

2. Tiny transition zones can be noticed between two neighboring zones. For instance, this is visible in Fig. 5.3 between the zones **0** and **1**, and between the zones **0** and **4**.

three studied waves. The objectives are to conduct mesh convergence analyses and to evaluate the influence of the non-conform refinement with various free-surface cell aspect ratio AR_{FS}. Table 5.5 gives a synthesis of the configurations.

(a) Uniform regular mesh `U64x64` (b) non-uniform regular mesh NU64x64

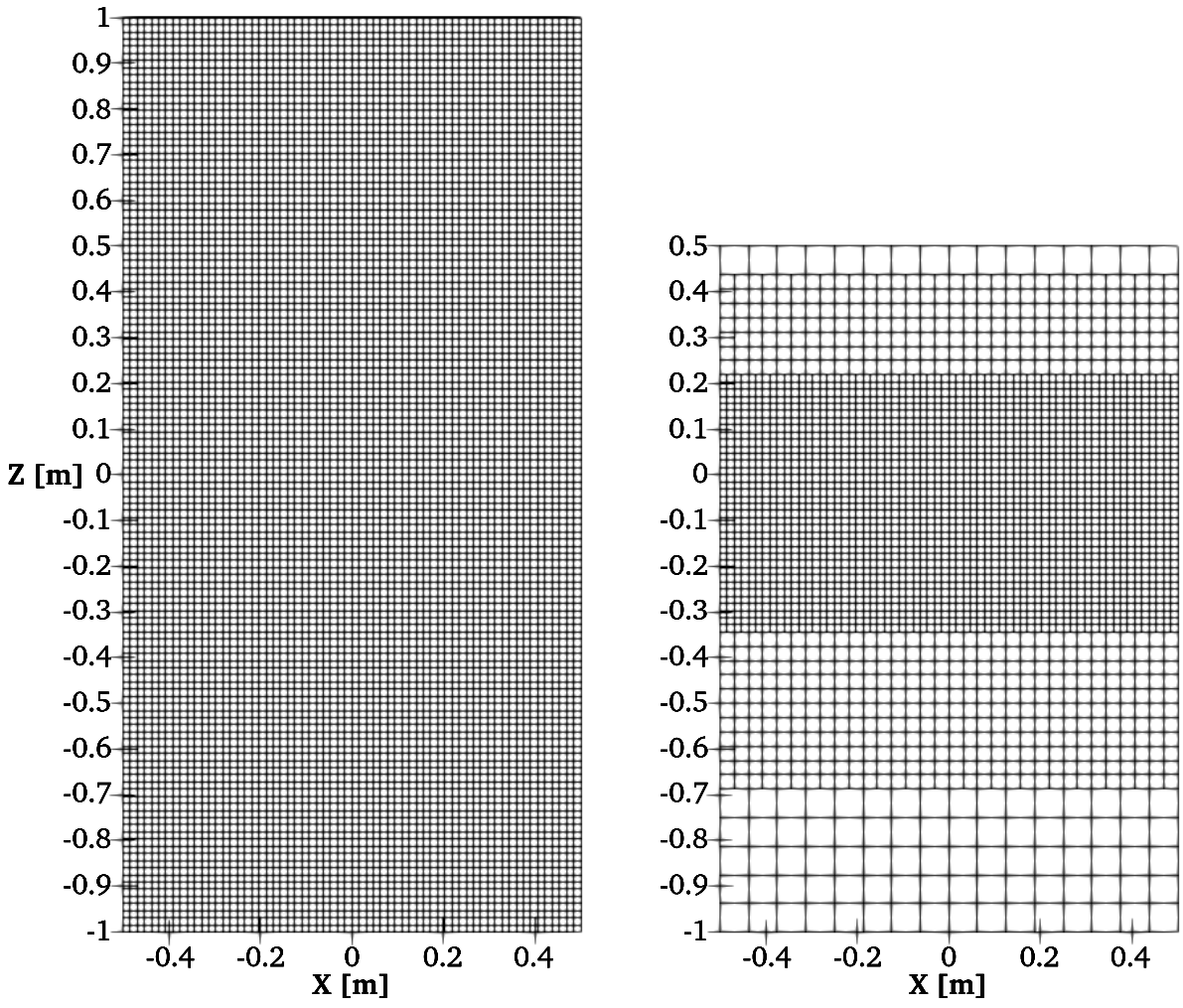


Figure 5.2 – Uniform and non-uniform mesh profiles used for periodic-wave propagation in periodic domain.

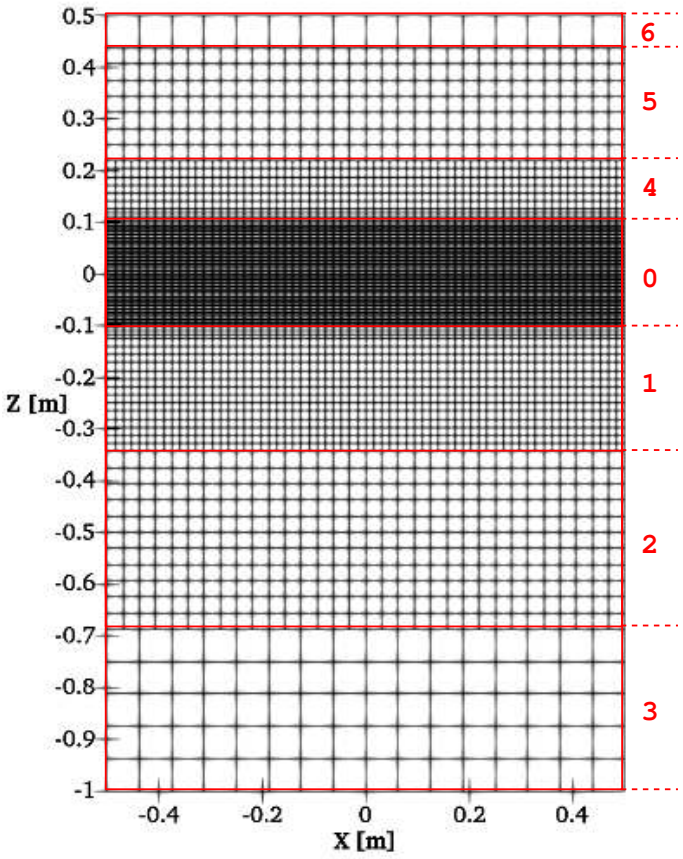


Figure 5.3 – Non-uniform stretched mesh `NU65x256` with refinement zones identification.

Table 5.4 – Characteristics of refinement zones used composing non-uniform meshes used for the regular wave propagation in periodic domain.

Zone index	zone size		Cell size	
	$z_{\min}$	$z_{\max}$	x-direction	z-direction
0	$-H$	H	Δx	Δz
1	$-\frac{1}{3}\lambda$	$-H$	Δx	Δx
2	$-\frac{2}{3}\lambda$	$-\frac{1}{3}\lambda$	$2\Delta x$	$2\Delta x$
3	$-\lambda$	$-\frac{2}{3}\lambda$	$4\Delta x$	$4\Delta x$
4	H	$\frac{1}{5}\lambda$	Δx	Δx
5	$\frac{1}{5}\lambda$	$\frac{2}{5}\lambda$	$2\Delta x$	$2\Delta x$
6	$\frac{2}{5}\lambda$	$\frac{1}{2}\lambda$	$4\Delta x$	$4\Delta x$

Table 5.5 – Studied configurations for the regular wave propagation in periodic domain.

Wave heights	$T/\Delta t$	**Mesh type**	**Mesh label**	$\lambda/\Delta x$	$\lambda/\Delta z$	$\mathrm{AR_{FS}}$
$H = 0.1$ m	{25, 50, 100, 100, 200, 400, 800, 1600}	Uniform regular	`U16x16`	16	16	1
			`U32x32`	32	32	
			`U64x64`	64	64	
			`U128x128`	128	128	
			`U256x256`	256	256	
$H = 0.1$ m $H = 0.05$ m $H = 0.01$ m	{25, 50, 100, 100, 200, 400}	Non-uniform regular	`NU16x16`	16	16	1
			`NU32x32`	32	32	
			`NU64x64`	64	64	
			`NU128x128`	128	128	
			`NU256x256`	256	256	
		Non-uniform stretched	`NU16x32`	16	32	2
			`NU16x64`		64	4
			`NU16x128`		128	8
			`NU16x256`		256	16
			`NU32x64`	32	64	2
			`NU32x128`		128	4
			`NU32x256`		256	8
			`NU32x512`		512	16
			`NU64x128`	64	128	2
			`NU64x256`		256	4
			`NU64x512`		512	8
			`NU128x256`	128	256	2
			`NU128x512`		512	4
			`NU256x512`	256	512	2

5.4 Steep regular wave propagation accuracy with backward and Crank-Nicolson MULES

The first analysis focuses on the accuracy of the newly implemented backward `MULES` compared to the Crank-Nicolson `MULES` on uniform regular meshes. The choice of uniform regular meshes is to avoid the errors and instabilities that non-uniform meshes could create. This first analysis is conducted on the steepest wave $H = 0.1$ m because the steepest the wave is, the more unstable and the less accurate the simulation tends to be (Choi et al., 2020). With isometric cells, the mesh is representative for this wave steepness in terms of number of cells per wave height and cells per wavelength with respect to the litterature.

As presented in Tab. 5.3, the tested time schemes are Crank-Nicolson with an off-centering coefficient of 1 (`CN1`) that corresponds to the standard Crank-Nicolson scheme, Crank-Nicolson with an off-centering coefficient of 0.95 (`CN95`) that is commonly assumed to be more stable and second-order backward (`BCK`). The tested `MULES` configurations are the standard one with a temporal and spacial first-order predictor and the `S-MULES` with the temporal high-order [3] and spatial first-order predictor. By default, when it is not specified, the standard `MULES` is used.

5.4.1 Field convergence

a) Wave elevation profile

First, the convergence of the simulation is controlled plotting the wave elevation profile along the longitudinal axis at $t = 10T$. Figure 5.4 shows these profiles obtained with a constant Courant number (Cfl ≈ 0.05) and several time and space resolutions. The analytical, `CN1`, `CN95`, and `BCK` results are presented with black dash-dotted line, solid line and dotted line, respectively. Only `BCK` were tested with $\Delta t = T/3200$ and $\Delta t = T/6400$. Qualitatively, it seems that the wave amplitude converges quicker towards the analytical solution than the phase shift. The latter remains noticeably far from the analytical one even with a small time step $\Delta t = T/6400$ and cell size $\Delta x = \lambda/1024$. Because of the computational cost, these fine parameters is not realistic for naval applications. In the literature, common values used for naval applications are $T/800 \leq \Delta t \leq T/200$ and $\lambda/200 \leq \Delta t \leq \lambda/50$. Besides, `CN1`, `CN95` and `BCK` provide very comparable results for each resolution used. This is all the more true comparing only `CN1` and `BCK` which are barely distinguishable from each other in Fig. 5.4. Finally a small

3. It is not formally second-order because Crank-Nicolson 0.95 is not (see Sec. 3.2 for more details)

distortion of the wave elevation profile is noticeable close to the crest with both CN1 and BCK for $\Delta = T/800$ (green lines).

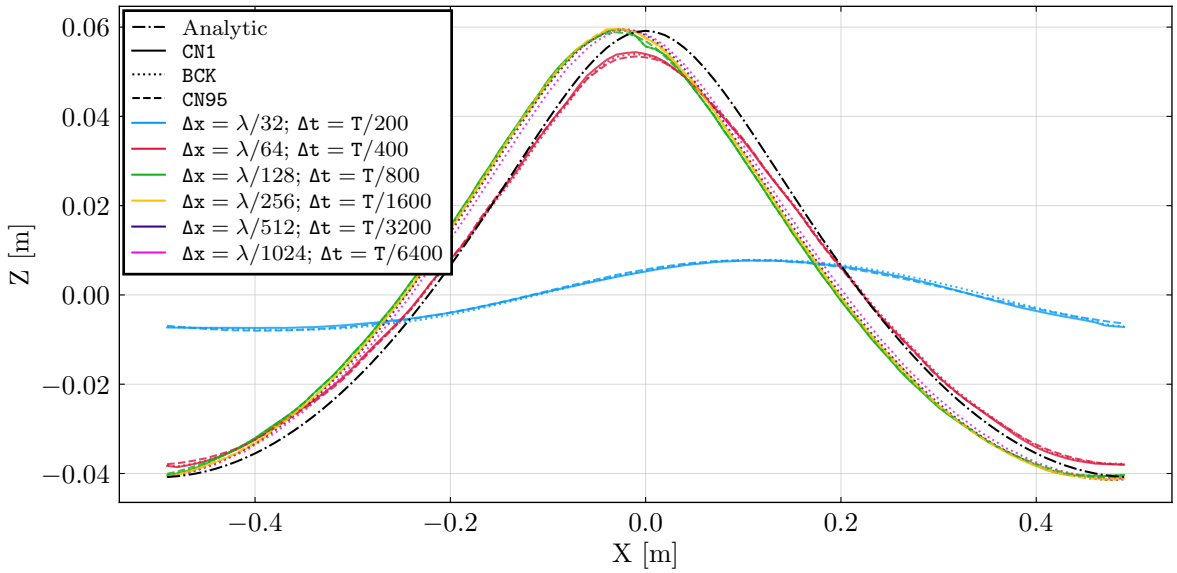


Figure 5.4 – Surface elevation profile with backward, Crank-Nicolson 0.95 and Crank-Nicolson 1 at $t = 10T$; depending on the discretization with $\mathrm{Cfl_x} \approx 0.05$; periodic wave H01; uniform mesh.

b) Velocity and pressure

Figure 5.5 and Figure 5.6 respectively show the horizontal velocity and dynamic pressure profiles obtained with CN95 (b) and BCK (c) at $t = 40T$ and the analytical field (a) obtained with potential stream-function theory used as the initial field at $t = 0\ s$. The mesh used for these simulation is U256x256 and the time step is $\Delta t = T/800$. On a qualitative point of view the fields seems to be well established without noticeable irregularities. CN95 and BCK time schemes provide a similar solution. No comparison can be shown using CN1 with the same configuration as the simulation failed after about forty periods.

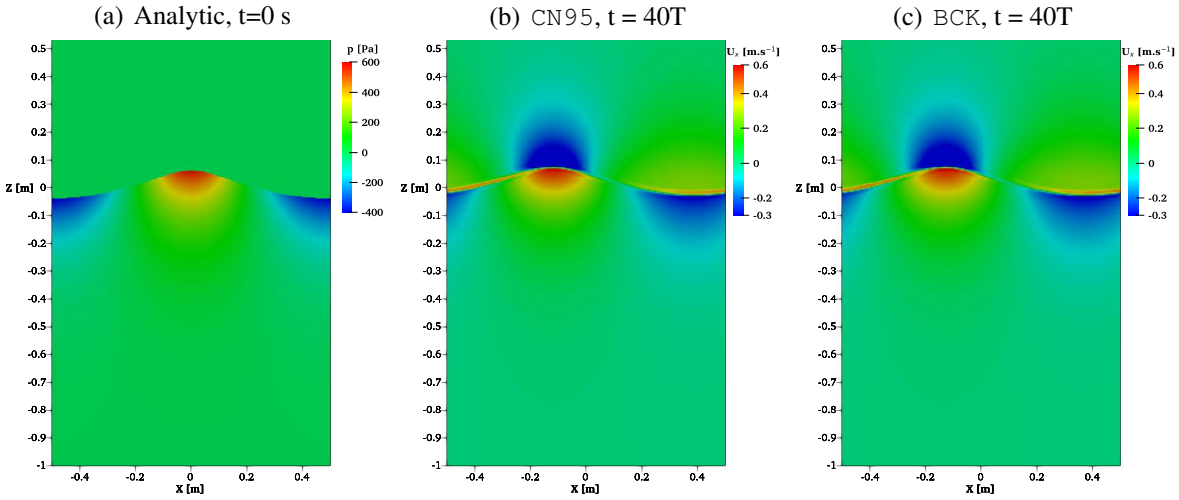


Figure 5.5 – Regular wave horizontal velocity field in periodic domain with uniform mesh `U256x256`; $\Delta t = T/800$.

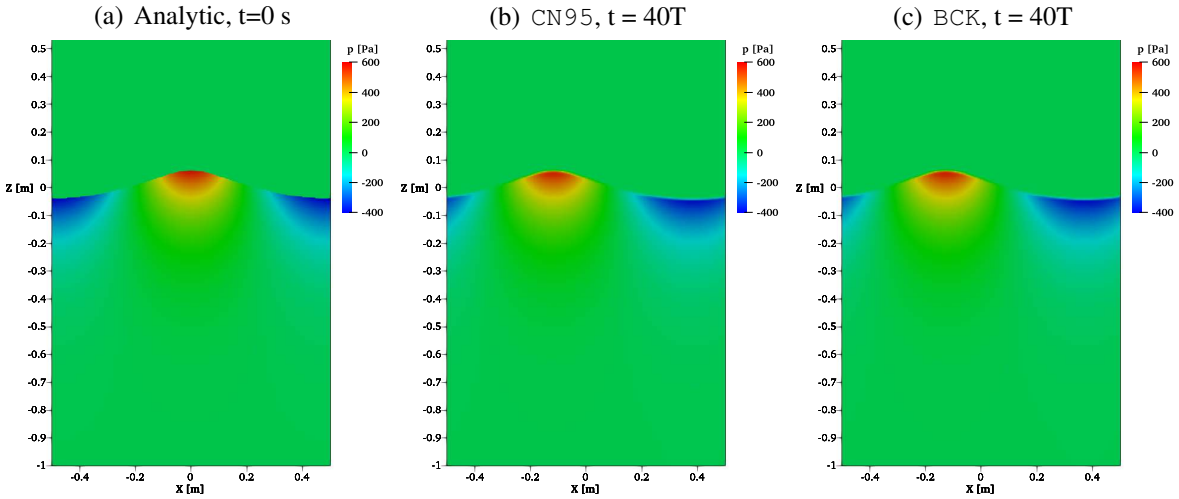


Figure 5.6 – Regular wave dynamic pressure field in periodic domain with uniform mesh `U256x256`; $\Delta t = T/800$.

A thin transition zone is noticeable across the air/water interface in Figure 5.5. As mentioned, the initial field is only computed in the water, with a single phase potential modelling, this is the reason why the velocity and dynamic pressure fields are null in the air at $t = 0$ s. The proposed potential flow solution is mono-fluid whereas the CFD model used in the present study is based on a single-field two-phase formulation with a compression term at the interface. The model is not developed to study this kind of interface flow but it imposes

continuity of the velocity field accross the interface and it is then natural that the solution implies a boundary layer between the phases. Figure 5.7 presents the horizontal velocity profile along the z-axis at $t = 10T$ and at the horizontal location of the crest of the wave, for the same set of simulations presented earlier. The convergence of the velocity in air is slower than what can be observed in the water and the solution is very different for the coarser mesh than for the other refinements, whatever is the time scheme used. This change appears between the resolution $\lambda/32$ and $\lambda/64$, which are grid resolutions not far from the typical values used in naval simulations. The horizontal velocity profile imposed as an initial field is plotted with bold black dash-dotted line. The initial unphysical null velocity in air is far from the converged value and a better initialization field could ease the convergence in the air. Another point that can be emphasized from Figure 5.7 is that the position of the top boundary seems sensitive because the solutions varies significantly around the boundary when changing the mesh. The solutions with the two most refined meshes show that the mesh `U256x256` can be considered converged.

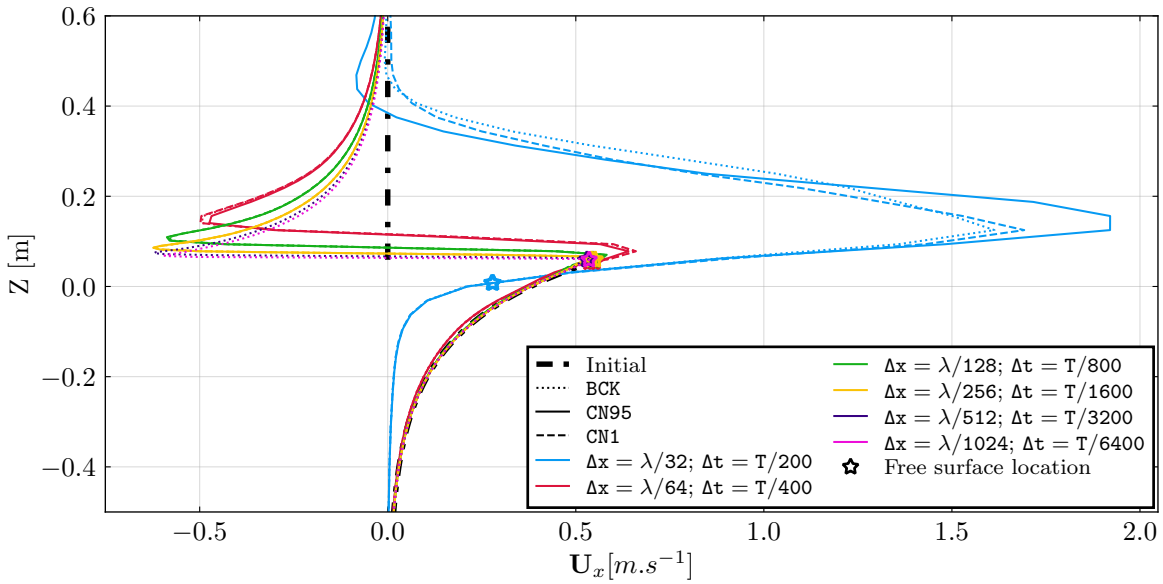


Figure 5.7 – Horizontal velocity profile with backward, Crank-Nicolson 0.95 and Crank-Nicolson 1 at $t = 10T$ and at the crest of the wave; depending on the discretization with $\mathrm{Cfl_x} \approx 0.05$; periodic wave `H01`; uniform mesh

In the following the focus is on the amplitude error (Sec. 5.1) obtained with second-order backward and Crank-Nicolson schemes. First, `BCK` is compared with `CN1` and then for robustness reason only the `CN95` scheme is considered. The same data are plotted under several views in order to identify multiple aspects of the numerical convergence.

5.4.2 Error in amplitude within the tested time and space resolution range with Crank-Nicolson 1 and backward time schemes

Figure 5.8 shows the absolute error of the first harmonic amplitude at $t = 37.5T$ depending on the Courant numbers in x-direction ($\mathrm{Cfl_x}$) obtained for various time steps Δt. The colors represent the time step and the marker styles represent the cell size. The solid lines connect the points obtained with `CN1` with a same Δt. The dotted lines do the same but with `BCK`. The logarithm scale is used on both the vertical and horizontal axis. The filled markers represent a positive error (overshoot of the analytic amplitude) when the empty markers represents a negative error (wave damping). This type of representation is frequently used in the following and is denoted by *iso-time-step view*.

All the meshes and time steps configuration tested with `BCK` are also tested with `CN1`. However, only the simulation reaching the targeted time $t = 40T$ are plotted. For the non-failed simulations with Crank-Nicolson 1, the results from `BCK` and `CN1` are close to each other. A large part of the simulations with `CN1` failed due to instabilities not well identified yet. Then, this scheme is assumed as not stable enough to be used for such simulations. Consequently, `CN95` is taken as reference scheme for the comparison with backward in the following. This choice of an off-centering coefficient below one is often done in the OpenFOAM literature. More detailed analysis of the the `BCK` results is done in the following.

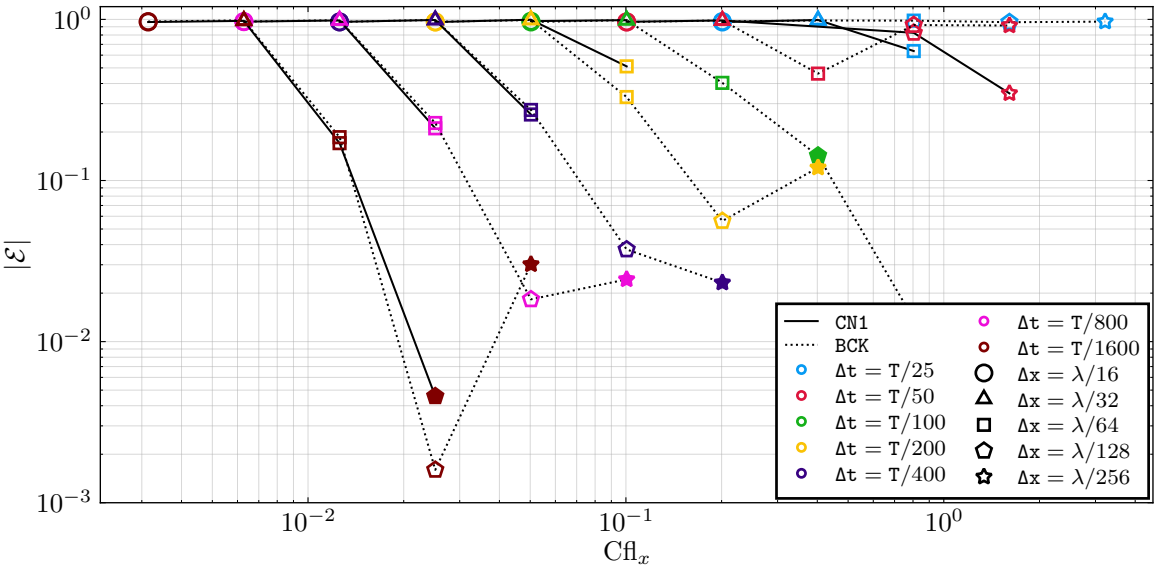


Figure 5.8 – *Iso-time-step view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number $\mathrm{Cfl_x}$ for various time steps Δt with backward and Crank-Nicolson; periodic wave `H01`; uniform mesh.

5.4.3 Error in amplitude within the tested time and space resolution range with Crank-Nicolson 0.95 and backward time schemes

a) Amplitude errors comparisons

Both Figure 5.9 and Figure 5.10 represent the absolute errors obtained with `BCK` and `CN95` within the tested range of time step and cell size. Fig. 5.9 is an *iso-time-step view* when Fig. 5.10 is an *iso-cell-size view*. In Fig. 5.9, the solid dark lines connect the Crank-Nicolson 0.95 data corresponding to a same time step when in Fig. 5.9, they connect the data corresponding to a same cell size. The same applies with the dark dotted lines and the backward scheme. These two *iso-time-step view* and *iso-cell-size view* representations highlight the spatial and temporal convergence, respectively. In these figures, two grey dashed lines represent the order-one and order-two slopes.

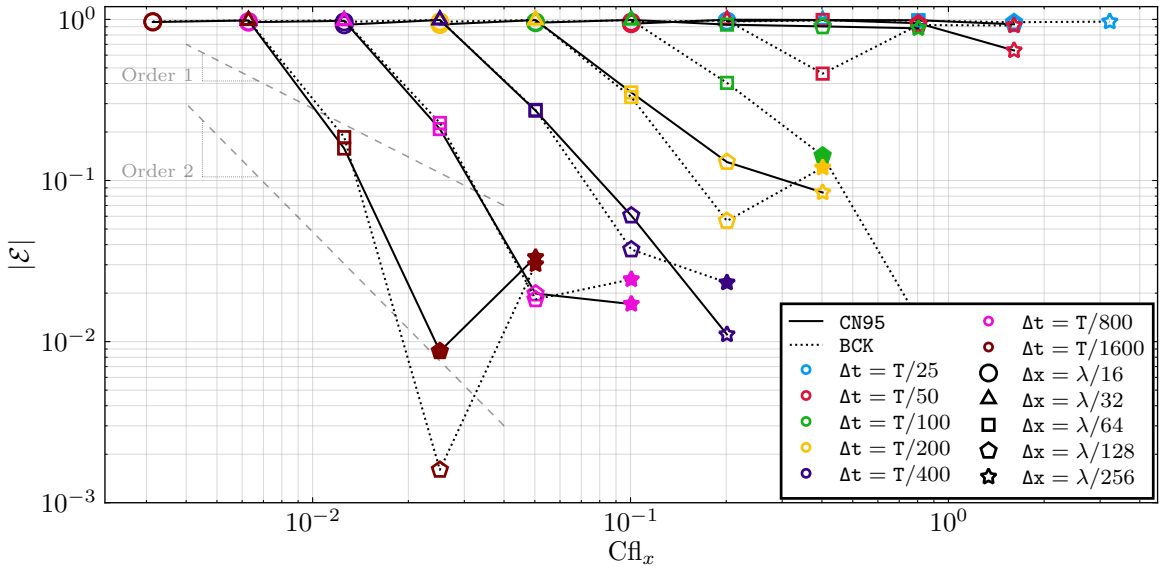


Figure 5.9 – *Iso-time-step view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number $\mathrm{Cfl_x}$ for various time steps Δt with backward and Crank-Nicolson 0.95; periodic wave `H01`; uniform mesh.

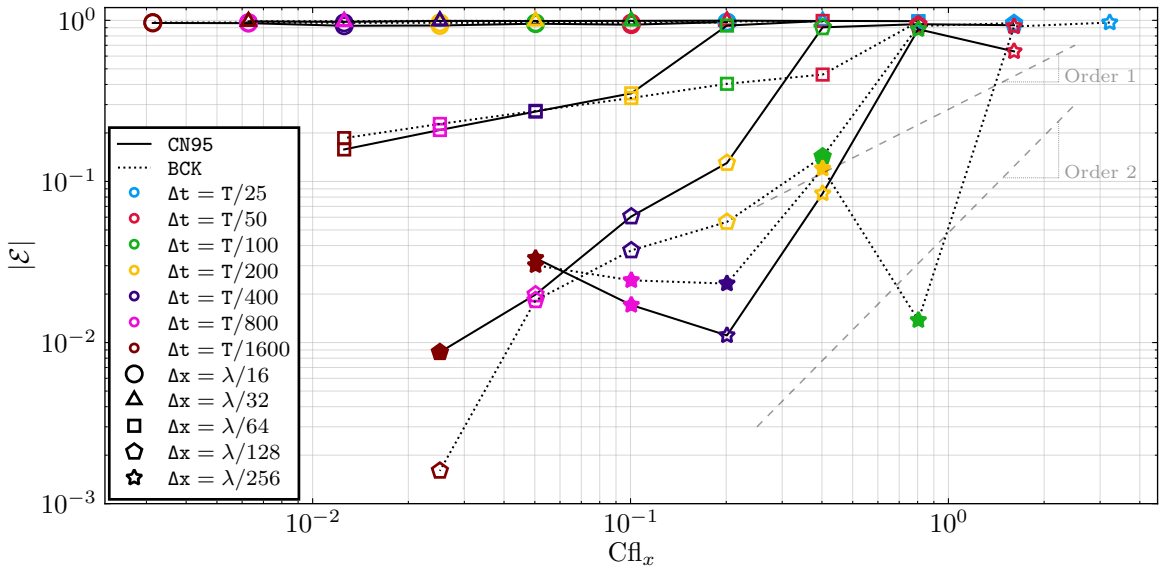


Figure 5.10 – *Iso-cell-size view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number $\mathrm{Cfl_x}$ for various cell sizes Δx with backward and Crank-Nicolson 0.95; periodic wave `H01`; uniform mesh.

Firstly, most simulations done with `CN95` reach $t = 40T$, indicating a significant

increase in robustness with respect to what was observed using `CN1`. With reaching $t = 40T$ as a robustness criterion, the stability of `BCK` and `CN95` is similar. Besides, some tested resolutions are too coarse to propagate the waves over forty periods because of the numerical damping. Using `CN95` gives 100% of error (flat free surface) for $\Delta t \geq T/100$ or $\Delta x \geq \lambda/32$. Using `BCK`, this 100% damping is reached for $\Delta t \geq T/25$ or $\Delta x \geq \lambda/32$. With `BCK` and $\Delta t = T/50$ the wave remains identifiable after forty periods but the free surface is deeply perturbated. This is consistent with the observations made in Figure 5.7 showing that a refinement $\Delta x = \lambda/32$ leads to a wrong air velocity field and a significant damping.

The general aspect of the convergence is more complicated than what was observed with the Taylor-Green test case. Looking at the `BCK` plots with time steps $\Delta t \leq T/200$ in Fig. 5.9, the error decreases faster than what it is expected with a second-order scheme and then increases back. This behavior is only noticeable using `CN95` for $\Delta t \leq T/800$. Visualizing the errors with the *iso-cell-size view* in Fig. 5.10 conducts to the same observations with a significantly non-monotonous temporal convergence with $\Delta x = \lambda/256$. With the finest meshes, the amplitude of the CFD wave tends to overshoot the analytic one. This phenomenon is likely to be the consequence of a convergence towards a result which is not the analytical one. This might be due to the bad accordance between the single-phase potential-flow initialization and the Navier-Stokes two-phase solution discussed earlier. Additionally to the initialization, the use of non energy-conservative numerical schemes might cause a gain of energy all along the simulation. When both the temporal and spatial refinement are fine enough, this gain of energy is not compensated by the numerical damping.

b) The energy gain

To verify the hypothesis of a gain of energy affecting the convergence, the time histories of both the kinetic (a) and potential energy (b) obtained with the finest mesh `U256x256` are plotted in Figure 5.11. Both `BCK` and `CN95` simulations gain energy over time with the smallest time step $\Delta t = T/800$. As the energy of a wave is directly correlated to the wave height, this observation is consistent with the hypothesis formulated in the comment of Fig. 5.9 and Fig. 5.10. This energy gain tends to converge toward a certain rate. This phenomenon explains the non-monotonic temporal convergence with the mesh `U256x256` in Fig. 5.10.

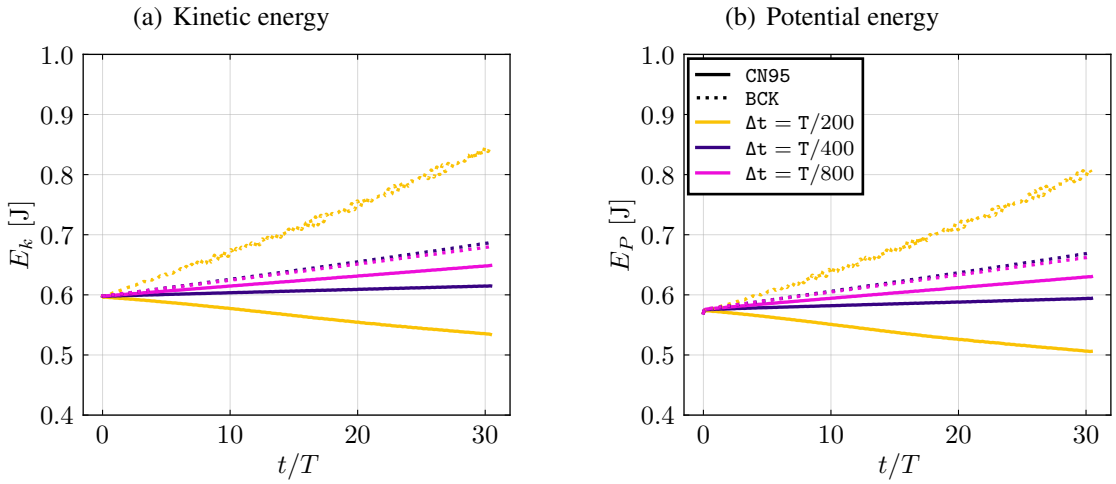


Figure 5.11 – Total kinetic and potential energy in the domain over time with backward and Crank-Nicolson 0.95 for several time steps; periodic wave `H01`; uniform mesh `U256x256`.

The problem of energy conservation solving the Navier-Stokes equations is well known. The use of collocated grid and face interpolation can lead to a non-conservative resolution in terms of energy (Ferziger and Peric, 2012) and it is quite hard to predict the evolution of this non-conservation (gain or loss). Some discretization methods ensure the kinetic energy conservation (Felten and Lund, 2006) but they are not used in this work. To go further in the analysis, the main origin of the gain of energy should be identified (flow of energy at the bottom and top boundaries, non-conservation due to face interpolations and PISO algorithm ...) and more refinements should be tested. This is not carried out in this document.

c) Convergence order verification

To remove the effect of the offset between the converged numerical and the analytical solution, the intrinsic errors are plotted in Figure 5.12 and Figure 5.13. The difference between these figures and Fig. 5.9 and Fig. 5.10 is the use of the finest discretization to evaluate $\mathscr{E}_f$. For a given plot in Fig. 5.12 corresponding to a given time scheme and time step, $\mathscr{E}_f$ is the error obtained with the same time scheme and time step but with the finest spatial discretization $\Delta x = \lambda/256$. Figure 5.13 shows the same data in *iso-cell-size view* where this time, the case with the smallest time step $\Delta t = T/1600$ is used as reference.

The use of intrinsic error improves the visualization of the convergence. However, the convergence order cannot be clearly identified. In Fig. 5.12, the slopes are between order 3 and 4 whereas the expected order is 2. In Fig. 5.12, the slopes are even less identifiable and seems

to be between order 1 and 2. With `BCK` and $\Delta x = 256$, the convergence is non-monotonous whereas with `CN95` it remains monotonous within the tested resolution range.

It appears that some phenomena influence significantly the convergence order and cause some overshooting of the wave elevation with some space and time resolution. In the following, to visualize the error regarding to the targeted value, and clearly expose overshoots, the intrinsic error is no longer analyses but only the error. As mentioned, plotting the absolute error with logarithm scale, the overshoot (positive error) are symbolized with a full color marker when the negative error are symbolized with an empty marker with colored edges. With this representation, a converging wave-elevation toward an higher value than the targeted one implies a fast decrease of the absolute error toward 0 followed by an increase up to the converged error.

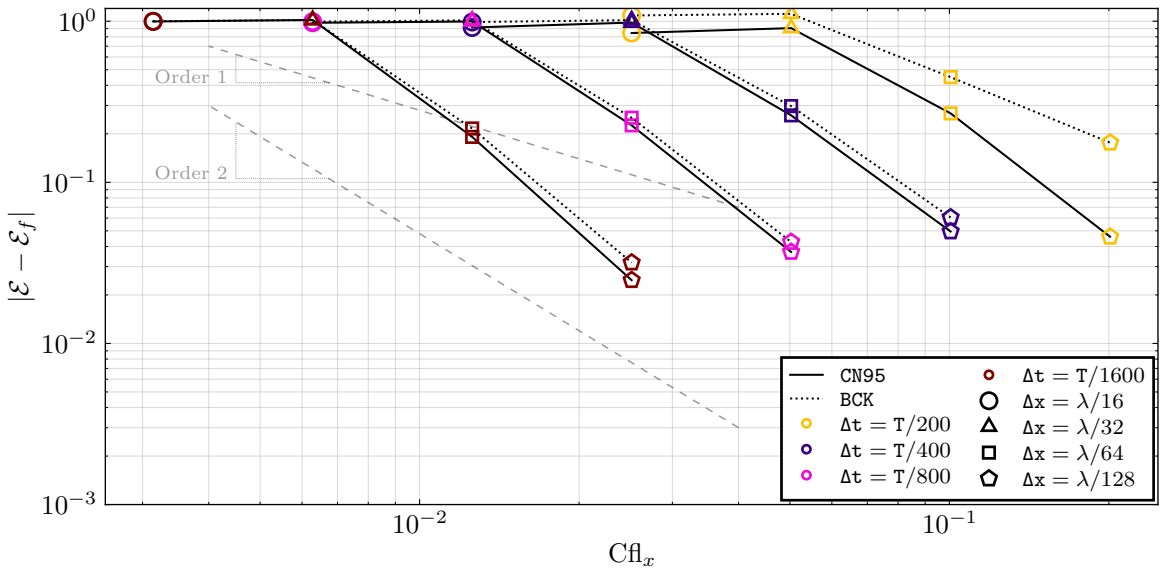


Figure 5.12 – *Iso-time-step view* of the surface-elevation first-harmonic amplitude intrinsic error at $t = 37.5T$ depending on Courant number Cfl_x for various time steps Δt with backward and Crank-Nicolson 0.95; periodic wave `H01`; uniform mesh.

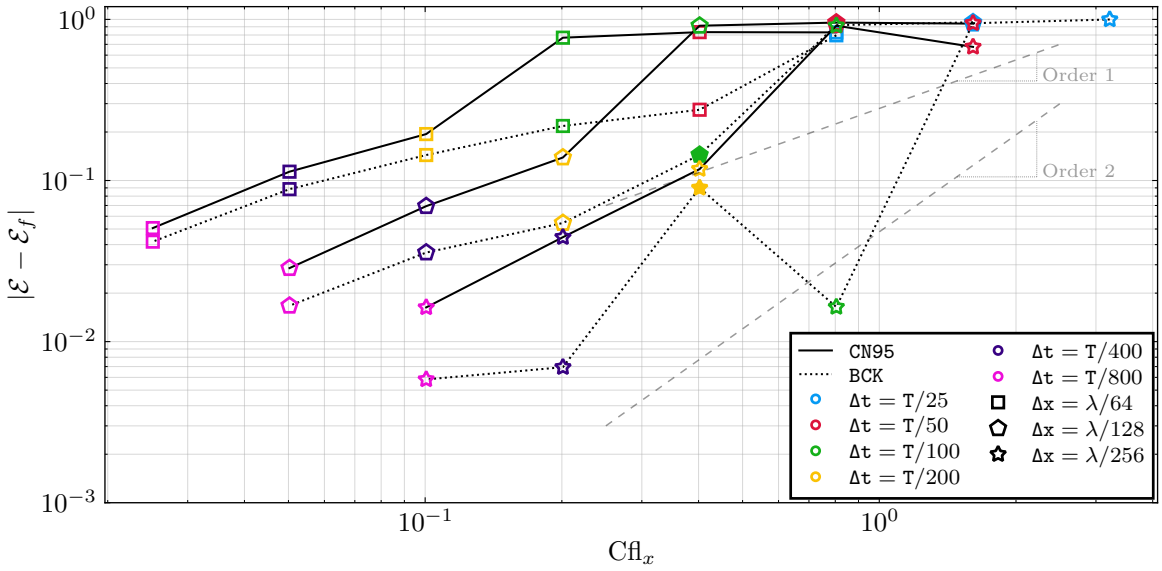


Figure 5.13 – *Iso-cell-size view* of the surface-elevation first-harmonic amplitude intrinsic error at $t = 37.5T$ depending on Courant number Cfl_x for various cell sizes Δx with backward and Crank-Nicolson 0.95; periodic wave `H01`; uniform mesh.

Plotting the error at fixed Courant number is a convenient way to verify the convergence order of the simulations. Figure 5.14 shows the evolution of the error over different cell refinements for two fixed Courant numbers, $\mathrm{Cfl}_x \approx 0.1$ (a) and $\mathrm{Cfl}_x \approx 0.2$ (b). The results with `CN95` are represented with a solid black line and those with `BCK` are represented with a dashed black line. Colors and marker styles represent as before specific time step and cell size, respectively. The slopes corresponding to order 1 and 2 are represented with dashed grey lines. With such a representation, both simulations with `CN95` and `BCK` behaves as second-order.

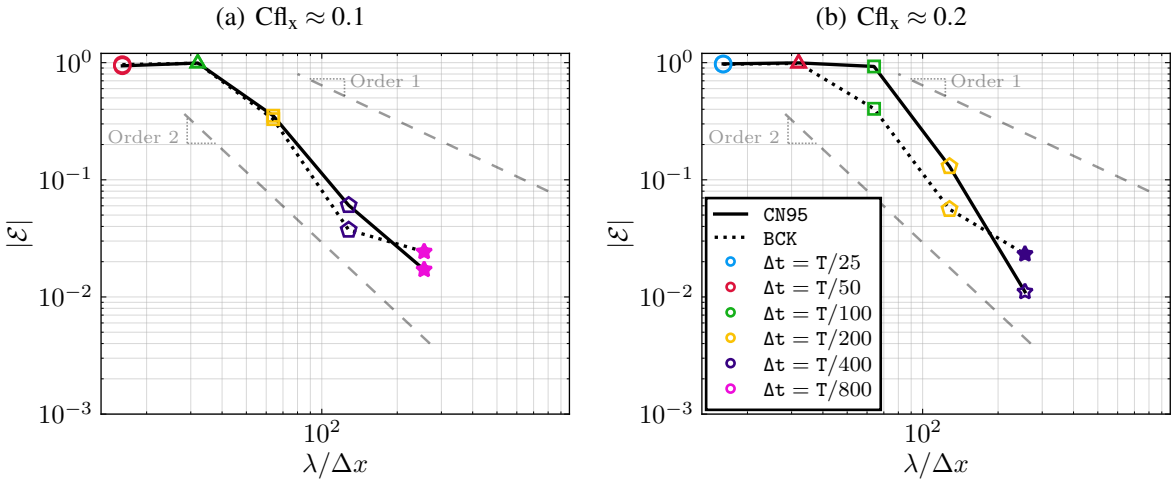


Figure 5.14 – Surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on cell discretization at fixed Courant number with backward and Crank-Nicolson 0.95; periodic wave `H01`; uniform mesh.

5.4.4 Spatial and temporal MULES

As detailed in Section 3.3.2, the `MULES` algorithm can be applied simultaneously on temporal and spatial schemes (`MULES`) or only on the spatial one (`S-MULES`). The Crank-Nicolson 0.95 `MULES` and Crank-Nicolson 0.95 `S-MULES` are compared using the *iso-time-step view* in Figure 5.15. Some `S-MULES` simulations are not represented as they have failed before reaching the targeted end. This indicates that `S-MULES` is less stable than the standard `MULES`. Besides, the completed `S-MULES` simulations give really close results to the ones obtained with `MULES` except for $\Delta t = T/100$ where `S-MULES` appears to be significantly more accurate.

The backward `MULES` and backward `S-MULES` are compared in Figure 5.16 using *iso-time-step view*. Unlike for `CN95`, `BCK S-MULES` has only encountered failures with really coarse temporal and spatial discretizations. However, the results between `MULES` and `S-MULES` significantly differ for some configurations. The phenomenon of getting higher accuracy with $\Delta x = \lambda/128$ than with $\Delta x = \lambda/256$ is amplified with `S-MULES` revealing possibly a faster energy gain with `S-MULES` than with `MULES`. Once again, this energy gain can be due to both the increasing of the non-conservative effect or the decreasing of the numerical damping. As the `MULES` algorithm was initially built for both spatial and time schemes during the VOF resolution, in the following the `S-MULES` version is abandoned.

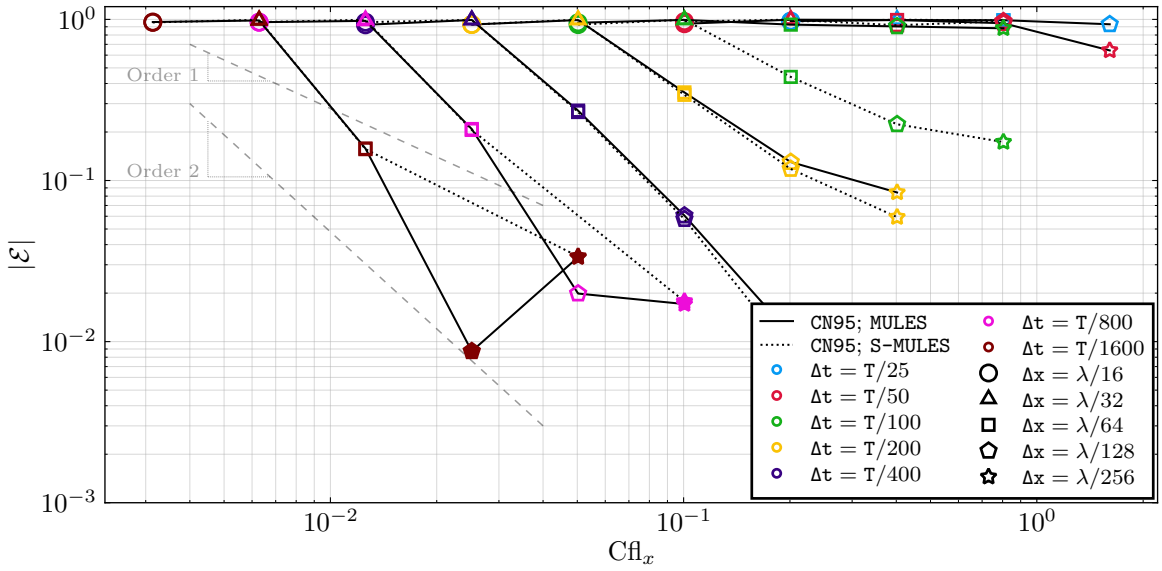


Figure 5.15 – *Iso-time-step view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number Cfl_x for various time steps Δt with Crank-Nicolson 0.95 MULES and S-MULES algorithms; periodic wave H01; uniform mesh.

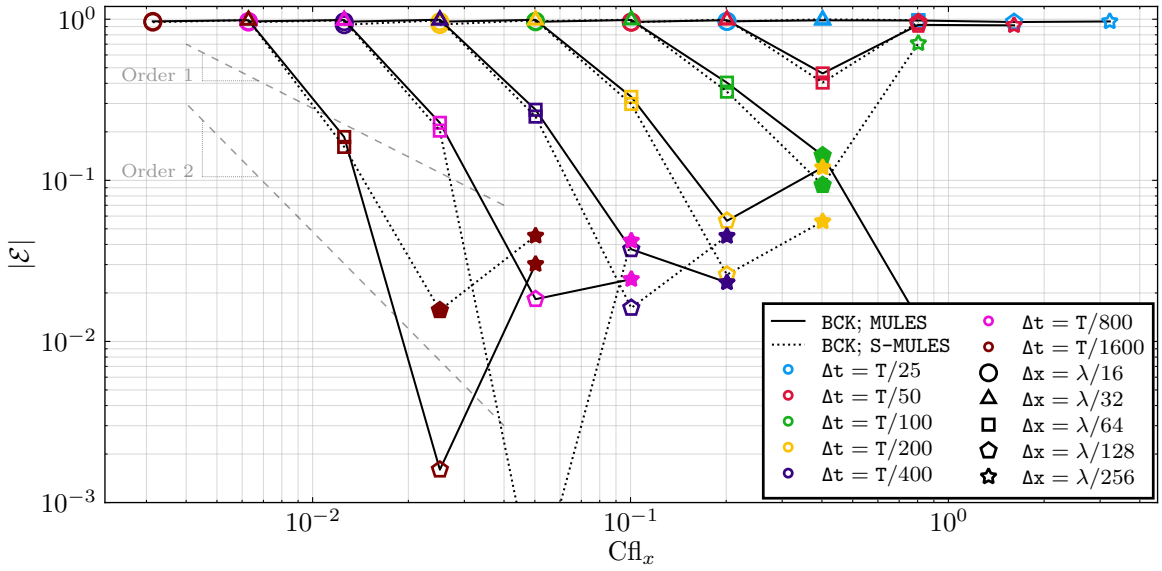


Figure 5.16 – *Iso-time-step view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number Cfl_x for various time steps Δt with backward MULES and S-MULES algorithms; periodic wave H01; uniform mesh.

5.4.5 Simulation with non-uniform mesh

In the work exposed in the previous paragraphs, the meshes were uniform with isometric cells. In a regular wave field, the magnitude of the velocity and the pressure decreases exponentially with the distance to the free surface. Consequently, in the literature and for naval industrial application, the meshes are progressively unrefined from the free surface to the bottom and top boundaries. The main interest of such a structure of mesh is to decrease the computational cost avoiding unnecessary fine zones. In this section, the simulations with non-uniform meshes presented in Section 5.3 are compared to the ones with uniform mesh.

a) Comparison of fields

Figure 5.17 shows the horizontal velocity map obtained with backward method with uniform mesh `U128x128` and non-uniform mesh `NU128x128` and $\Delta t = T/400$. The use of a non-uniform mesh with smaller air domain does not create visible spurious velocities and both configuration seem to provide a well established field. However, even though the smaller density of air leads to a smaller energy in the air field, it can be seen in Figure 5.17 that velocity gradients are present high in the air part where a degraded resolution is used. In the present case, the used non-uniform mesh is not unrefined symmetrically in the vertical direction (see Fig. 5.3) The use of the present version of non-uniform meshes could lead to some disturbances at the free surface and finally impair the simulations. Using a symmetric refinement between air and water could be beneficial and subject of a dedicated study.

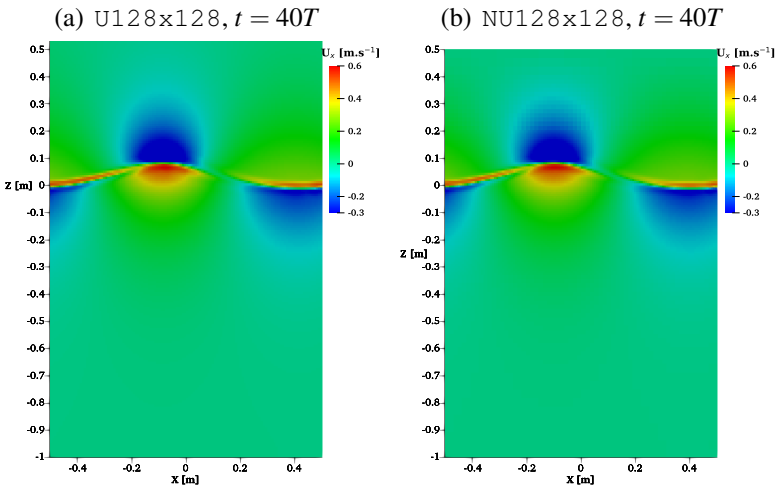


Figure 5.17 – Regular wave horizontal velocity field in periodic domain with uniform and non-uniform meshes `U128x128` and `NU128x128`; backward `MULES`; $\Delta t = T/400$.

b) Comparison of the wave amplitude error

Figure 5.18 shows an *iso-time-step view* of the errors obtained with `CN95` on uniform and non-uniform meshes with the wave `H01`. Results between uniform and non-uniform meshes are really close, the accuracy is not significantly impacted by the change in the mesh structure.

Figure 5.19 shows the same analyses but using `BCK`. In this case the use of a non-uniform mesh tends to change the error amplifying the overshoot of the targeting wave elevation. This is particularly visible with $\Delta t = T/200$ and $\Delta t = T/400$.

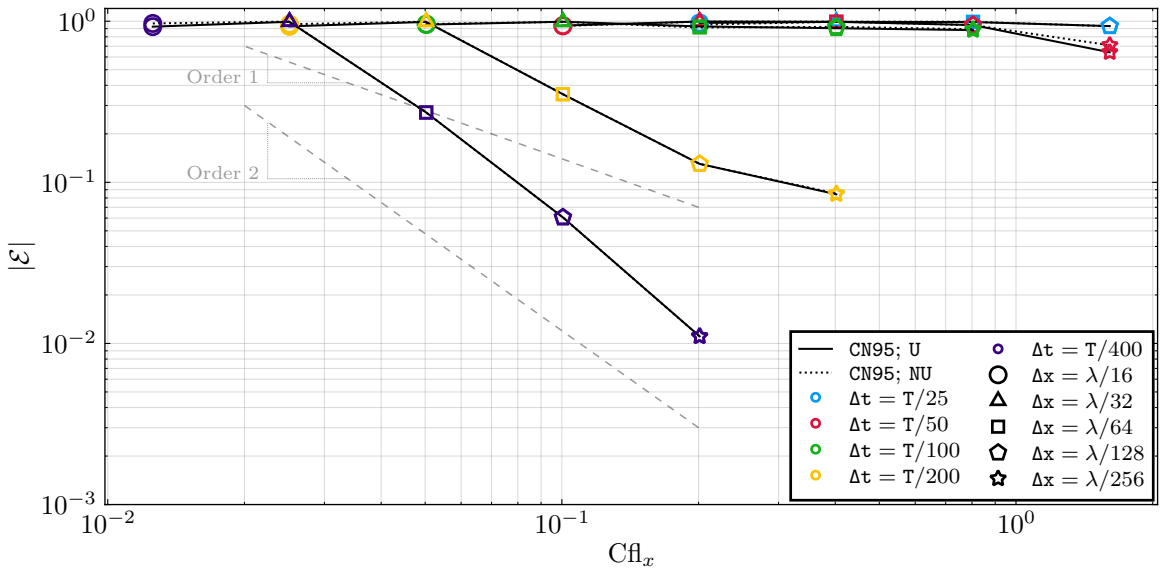


Figure 5.18 – *Iso-time-step view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number Cfl_x for various time steps Δt for CrankNicolson 0.95 with Uniform mesh and non-uniform refined mesh; periodic wave `H01`; uniform mesh.

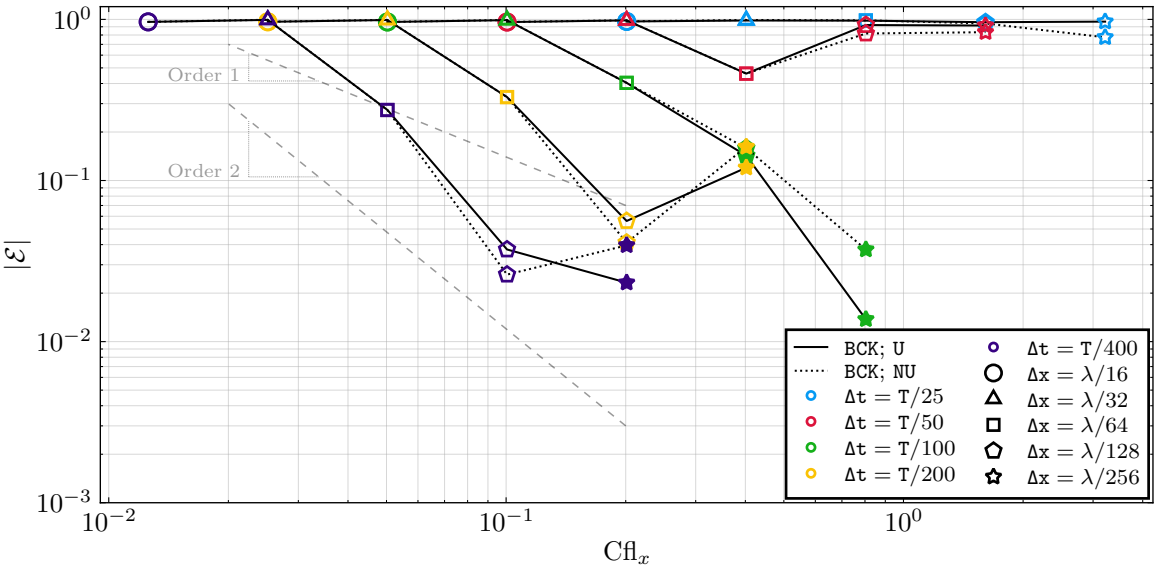


Figure 5.19 – *Iso-time-step view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number Cfl_x for various time steps Δt for backward with uniform and non-uniform mesh; periodic wave `H01`; $\mathrm{AR}_{\mathrm{FS}}$=1.

Even if the use of non-uniform mesh might slightly negatively impact the accuracy using backward scheme, the computational cost benefit of non-uniform mesh largely justifies its use for industrial naval applications. Consequently, in the following non-uniform refinement as defined in Table 5.4 are used.

5.5 Regular wave propagation accuracy depending on the wave steepness and the cell aspect ratio

For the steepest wave, the efficiency of non-uniform meshes built with the method described in Section 5.3 was shown. The next step is to study the influence of cell aspect ratio at the free surface (`ZONE 0`) for various wave steepnesses.

5.5.1 Optimal free-surface aspect ratio

In the present case, as seen with the steepest wave, verifying the numerical errors is conveniently done analyzing convergences at fixed Courant numbers. Figure 5.20 shows the evolution of the error depending on cell size for each studied steepness. Two ratio between time

step and cell-size are plotted. $\frac{\Delta t}{\Delta x} = \frac{128}{400}$ in (a), (c), (e) and $\frac{\Delta t}{\Delta x} = \frac{128}{200}$ in (b), (d), (f). For a given wave height it corresponds to two distinct Courant numbers. These $\mathrm{Cfl_x}$ are different from a wave height to another because of a distinct reference velocity (Eq. 5.1). In these figures the Crank-Nicolson 0.95 and backward time schemes are represented with solid and dotted lines, respectively. A given colors is assigned to a given free-surface cell aspect ratio $\mathrm{AR_{FS}}$. The markers shapes keep the same signification as in this whole chapter, representing the cell sizes.

In Fig. 5.20, the slopes of the plots are between first and second-order. The order seems to drop as the wave heigth/steepness decreases. A significant impact of the free-surface cell aspect ratio is seen. For a given wave height, an optimal $\mathrm{AR_{FS}}$ provides errors in accordance with the second-order behavior.

The key point here is that the most accurate aspect ratio depends on the steepness. Table 5.6 synthesizes the optimal $\mathrm{AR_{FS}}$ identified using Fig. 5.20. The choice between `CN95` and `BCK` scheme does not change these optimal free-surface cell aspect ratios.

Because of the choice made to limit the vertical cell size to $\Delta z = \lambda/512$, only few simulations are done using $\mathrm{AR_{FS}} = 8$. To complete the results, thinner vertical refinements should be tested.

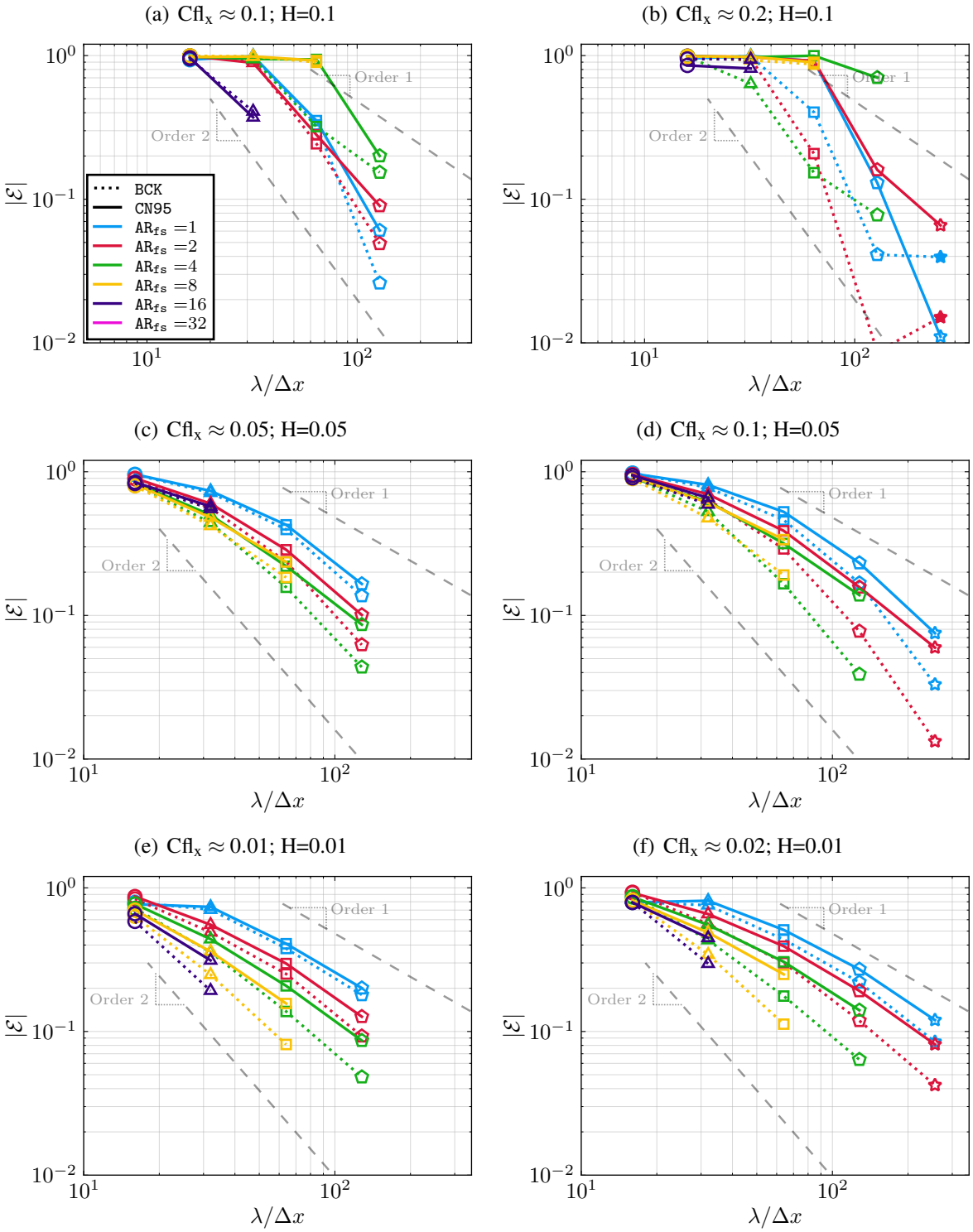


Figure 5.20 – Surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on cell aspect ratio at free-surface (AR_{FS}) at various Courant number Cfl_x for backward and Crank-Nicolson 0.95; periodic wave; non-uniform mesh.

Table 5.6 – Optimal aspect ratio for various wave steepness.

Wave	Steepness	Optimal aspect ratio
H001	1%	8 or 16
H005	5%	4 or 8
H01	10%	1 or 2

5.5.2 Comparison of the errors with backward and Crank-Nicolson 0.95 schemes

To compare the errors get using backward and Crank-Nicolson 0.95, *iso-time-step views* are plotted in Figure 5.21, Figure 5.22 and Figure 5.23 for the wave heights H=0.1, H=0.05, H=0.01, respectively. For each wave, AR_{FS} is the one prescribed previously (Tab. 5.6).

In each figure, The `BCK` provides more accurate results than `CN95`. For the steepest wave (Fig. 5.21), the results were already presented in Section 5.4.

Then, it can be noticed that the higher the steepness, the higher the error for low resolutions and the higher the apparent convergence order. Once again, finer resolution, other mesh structures and energy gain control could explain these observations.

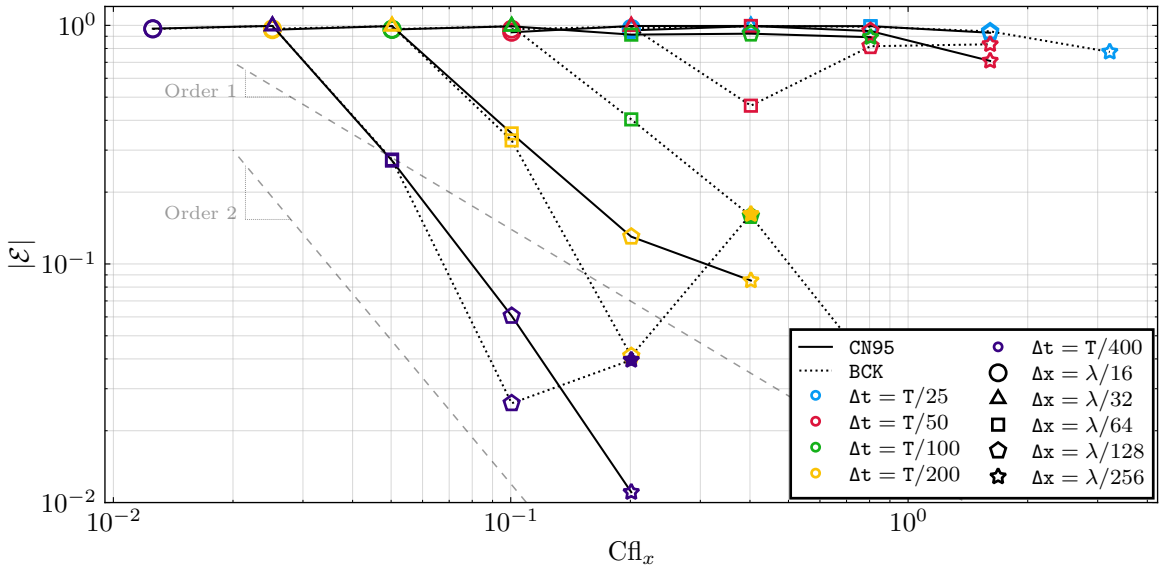


Figure 5.21 – *Iso-time-step view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number Cfl_x for various time steps Δt with backward and Crank-Nicolson 0.95; periodic wave `H01`; uniform mesh; AR_{FS}=1.

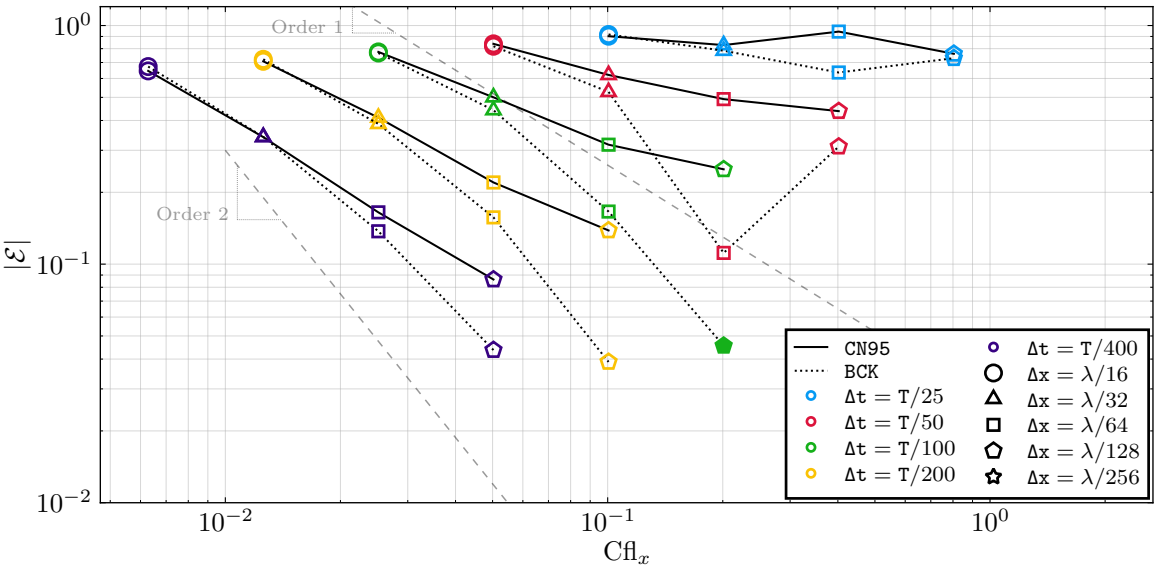


Figure 5.22 – *Iso-time-step view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number $\mathrm{Cfl_x}$ for various time steps Δt with backward and Crank-Nicolson 0.95; periodic wave `H005`; uniform mesh; $\mathrm{AR_{FS}}$=4.

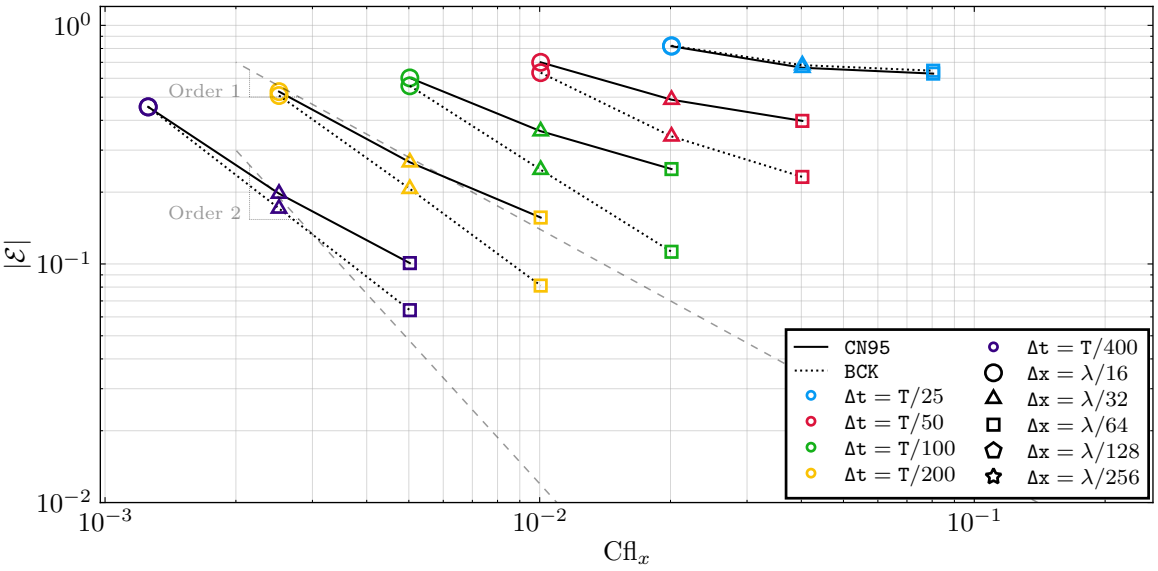


Figure 5.23 – *Iso-time-step view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number $\mathrm{Cfl_x}$ for various time steps Δt with backward and Crank-Nicolson 0.95; periodic wave `H001`; uniform mesh; $\mathrm{AR_{FS}}$=8.

5.6 Comparison of the efficiency of various discretizations

The last point to analyze is the efficiency. Figures 5.24, 5.25 and 5.26 show *iso-cell-size views* of the error but using the CPU time for the abscissa rather than the Courant number. The logarithm scale is still used. The figures concern the waves `H01`, `H005` and `H001`, respectively. The meshes respect the optimal AR_{FS} previously identified.

In these figures, taking into account the computational cost (CPU time), it appears that for a same spatial and temporal resolution, both the simulation using `BCK` and `CN95` are achieved with a similar CPU time. However, there are some exceptions for which the backward scheme induced an over-cost. For instance, in Fig. 5.24, with $\Delta t = T/200$ and $\Delta x = \lambda/256$ (yellow star), the simulation with `BCK` is significantly less accurate and slower than the one with `CN95`. This loss of efficiency could be linked to a field perturbation caused by a non well-design mesh or numerical set-up. The problem of the resolution of the air phase exposed in Sec. 5.4 Sec. 5.4.1 and Sec. 5.4.5 is potentially involved. Some other points show a similar loss of efficiency using backward especially with the finest resolution. Except for these local issues, the backward scheme shows a clear global improvement of the efficiency compared to the Crank-Nicolson 0.95 time scheme.

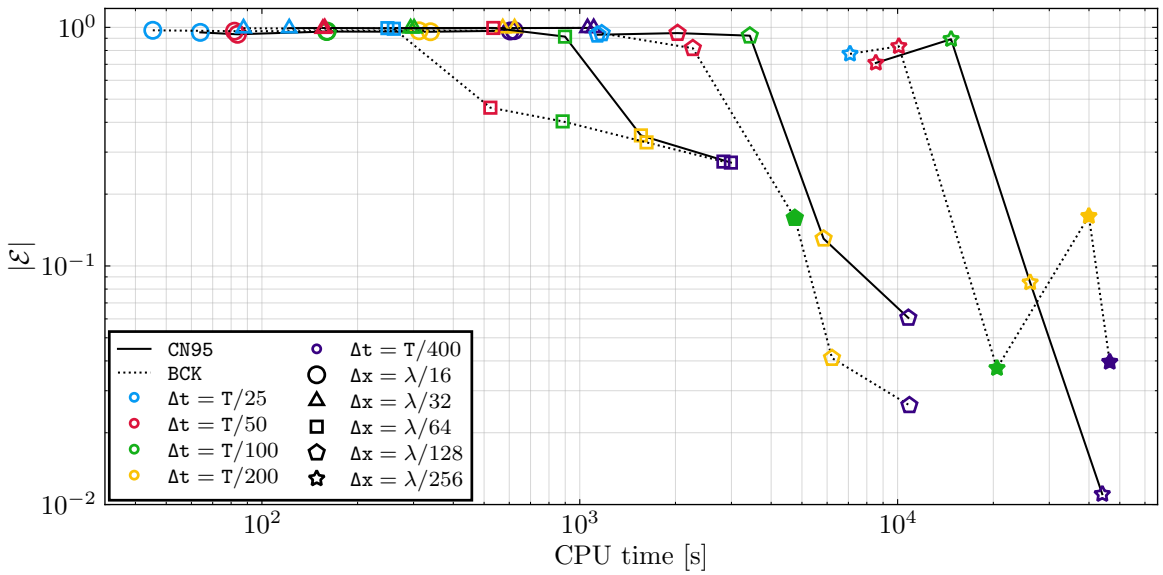


Figure 5.24 – *Iso-cell-size view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ and CPU time for various cell sizes Δx with backward and Crank-Nicolson 0.95; periodic wave `H01`; non-uniform mesh; AR_{FS}=1.

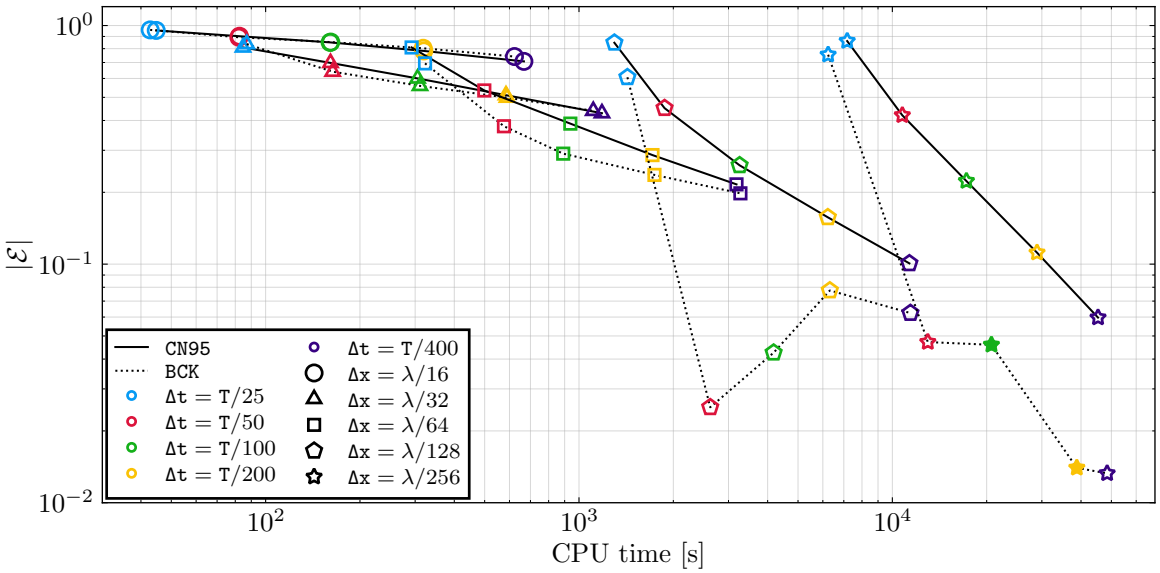


Figure 5.25 – *Iso-cell-size view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ and CPU time for various cell sizes Δx with backward and Crank-Nicolson 0.95; periodic wave `H005`; non-uniform mesh; AR_{FS}=4.

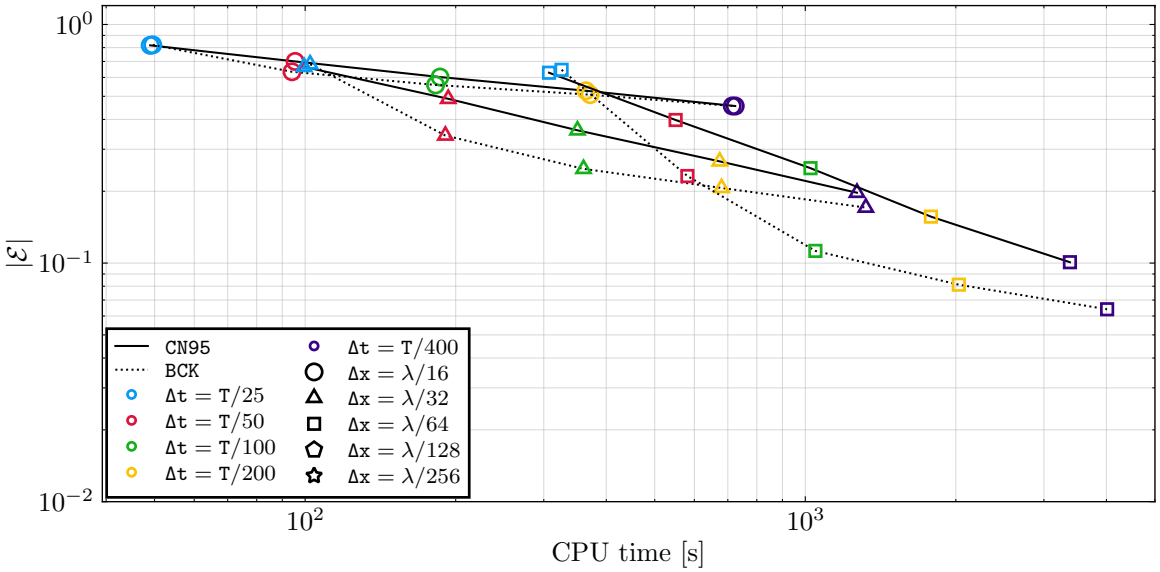


Figure 5.26 – *Iso-cell-size view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ and CPU time for various cell sizes Δx with backward and Crank-Nicolson 0.95; periodic wave `H001`; non-uniform mesh; AR_{FS}=8.

Finally, Figure 5.27 shows the impact of the AR_{FS} choice. In this figure, the results

are from the simulations of wave H001. Meshes with $AR_{FS} = 4$ and $AR_{FS} = 8$ are compared. For this wave, the previously identified optimal AR_{FS} in terms of accuracy is 8. On one hand, increasing the vertical refinement (reducing the cell size Δz) implies an increase of the number of cells and therefore an increase of the computational cost. On the other hand, for a given space and time refinement, the accuracy with $AR_{FS} = 8$ is higher than with $AR_{FS} = 4$. Fig. 5.27 shows that for this type of simulation the choice of $AR_{FS} = 8$ is still more efficient and the gain of accuracy compensates the small increase of computation cost. For a same error, using a convenient free-surface cell aspect ratio leads to a reduction of the computational cost by a factor 3.

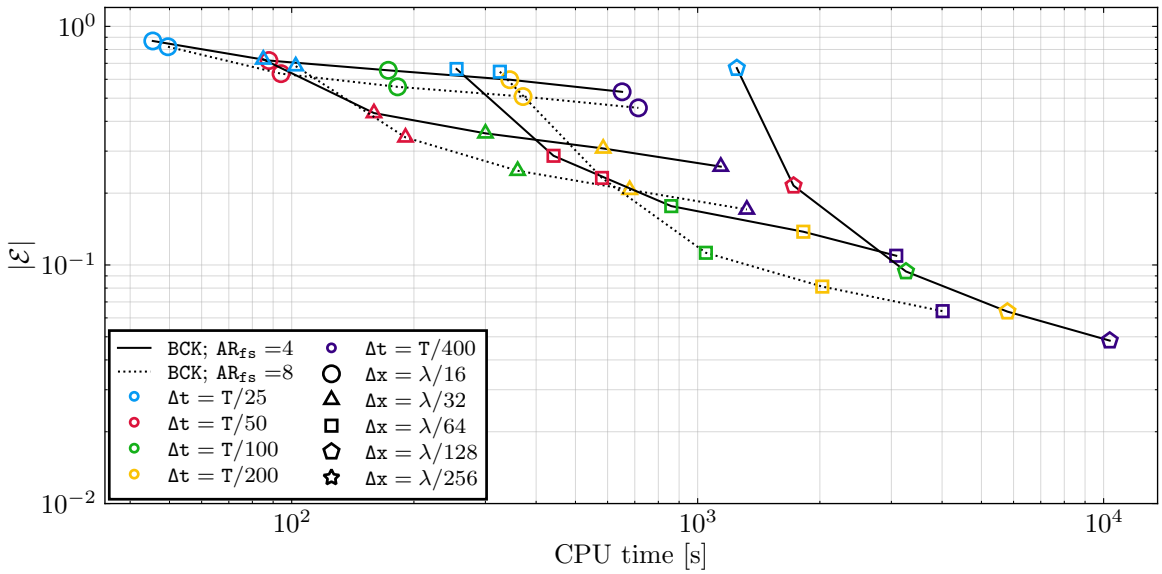


Figure 5.27 – *Iso-cell-size view* of the surface-elevation first-harmonic amplitude error at $t = 37.5T$ and CPU time for various cell sizes Δx with AR_{FS}=4 and AR_{FS}=8; periodic wave H001; non-uniform mesh; backward.

5.7 Conclusion

In this chapter, it was shown that with *foamStar*, both Crank-Nicolson 0.95 and backward time scheme are accurate for approximately second-order simulations of regular waves propagating in periodic bi-dimensional domain with various wave steepnesses, 1%, 5% and 10%. Within a range of time and space resolution commonly used for naval simulations, the MULES implementation of second-order backward scheme is slightly more accurate and efficient than the Crank-Nicolson 0.95 MULES.

Some recommendations have been made for the structure of meshes. Firstly, the mesh structure established in Sec. 5.3 is convenient for efficient simulations. Then, a strong correlation between optimal free-surface cell aspect ratios and wave steepnesses have been identified. About a factor 5 to 10 between this aspect ratio and the wave steepness seems to be convenient. For a given regular wave with a wavelength λ and a wave height H, an optimal aspect ratio can be written as

$$\frac{\lambda}{5H} \leq {\mathrm{AR}_{\mathrm{FS}}}^{Opti} \leq \frac{\lambda}{10H}. \tag{5.3}$$

CHAPTER 6

WAVE PROPAGATION WITH RELAXATION ZONES

The next step toward a proper naval simulation is the use of realistic domains, several wavelengths long, with relaxation zones and RANS model. The study presented in this chapter is in line with the work done in the Numerical Wave Tank (NWT) working group of the Reproducible CFD Modeling Practice for Offshore Applications Joint Industrial Project (JIP) led by Technip Energies.

This work is the continuation of the previous section where periodic waves were studied. The same three wave conditions are studied with the same meshing method. The difference lies in the domain configuration and the wave generation. In the previous section, the wave fields were set at the initialization and then freely propagated in a periodic domain. In this chapter, waves are generated and absorbed using relaxation zones. In order to progressively get closer to the naval simulations, the first configuration studied here uses the Navier-Stokes model (NS) as in the previous chapter. Then, a turbulent RANS model is introduced and finally simulations are perfomed in a moving frame as it is done for the case of a moving ship. The following analyses do not go deep into the Verification procedure. The objectives are to identify some potential issues induced by the introduced features and to build recommendations on how to use the current *foamStar* implementation to perform naval simulations in waves.

Remark: *The following work is done using a modified foamStar version in witch the relaxation zone updates are organized differently compared to the flow chart presented in Figure 3.10. For the version used in the following the velocity update is done between the "u estimation step" and the PISO loop start.*

6.1 Case definition

The numerical domain configuration and the wave characteristics are defined according to the JIP procedure (Bouscasse et al., 2021). The characteristics of the simulated waves are

recalled in Table 6.1. As for the periodic waves study, in the following the labels `H01`, `H005` and `H001` refer to the selected waves with the wave heights $H = 0.1$ m, $H = 0.05$ m and $H = 0.01$ m, respectively. Those waves are the same as the previously used ones (Sec. 5). The stream theory model implemented in *foamStar* is used in the relaxation zones to impose the wave fields [1].

Two domain configurations are used in this section. One with an inlet and outlet relaxation zone and another with only an outlet relaxation zone. Figure 6.1 shows the first configuration. The size of the relaxation zones follows the recommendations established by Choi et al. (2020). The inlet relaxation zone is 1λ long and the outlet one is 2λ long. The so-called pure CFD domain is 7λ long and centered at the position $x = 0$. This domain configuration is denoted by `RZi1o2`. Figure 6.2 shows the second configuration for which the relaxation zone is only located at the outlet [2]. The outlet relaxation zone is 3λ long and the pure-CFD-domain is 7λ long with a central position $x = -\lambda$. This domain configuration is denoted by `RZo3`.

As indicated in Fig. 6.1 and Fig. 6.2, for both configurations the wave direction is toward positive x-direction (orange arrow) and the frame velocity is toward negative x-direction (white arrow). The moving frame direction corresponds to a simulation of a boat moving with forward speed in head waves.

For applications such as the added resistance in head waves, the simulation domain geometry is often not based on the wavelength but on the ship length (L_{PP}) in order to solve the diffracted field accurately. In the literature (see e.g. Simonsen et al., 2013; Kim et al., 2017; Mofidi et al., 2018) a standard recommendation is to define a domain length of about 4 to 6 L_{PP}. One example of standard naval domain can be found in Chapter 7. Besides, wavelengths between 0.5 L_{PP} and 2 L_{PP} are commonly encountered in naval simulations (Fujii (1975), Larsson et al. (2010), Larsson et al. (2018)...). Consequently, the domain configurations studied in this section are relevant to naval simulations with short wavelengths and a fictive scale model boat for which $L_{PP} \approx 2$ m.

In the following, in order to evaluate a longitudinal Courant number $\mathrm{Cfl_x}$, the reference velocity for a given wave is still:

$$\mathbf{u}_{\mathrm{ref}} = \frac{\pi H}{T}. \tag{6.1}$$

1. for more detail see Sec. 2.3.1
2. The reasons behind this choice are detailed in the following.

Table 6.1 – Regular wave characteristics.

Item	**Unit**	`H001`	`H005`	`H01`
Depth (D)	[m]	1	1	1
Wavelength (λ)	[m]	1	1	1
Wave period (T)	[s]	0.80	0.79	0.76
Wave height (H)	[m]	0.01	0.05	0.1
Wave steepness ($\varepsilon = H/\lambda$)		1 %	5 %	10 %

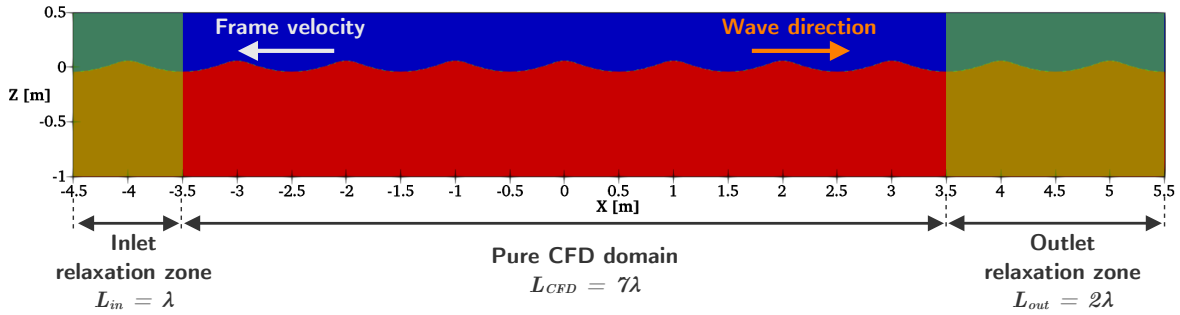


Figure 6.1 – 2D domain dimensions used for pure wave propagation with Navier-Stokes model. (`RZi1o2`)

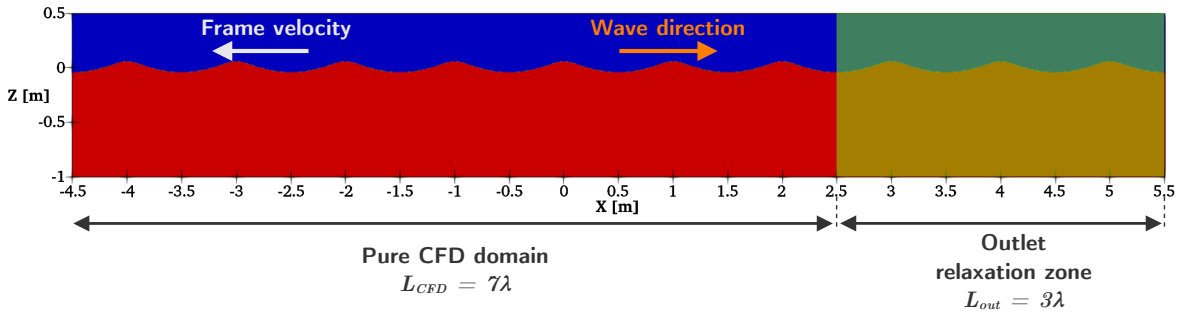


Figure 6.2 – 2D domain dimensions used for pure wave propagation with RANS model. (`RZo3`)

The boundary conditions in `RZi1o2` and `RZo3` are synthesized in Table 6.2. For more details on these BCs see Section 3.3.6. These BCs are used for both NS and RANS simulations done along this study. For NS simulations modelling of the turbulence is not taken into account, so the turbulent fields `k`, `omega` and `nut` are not activated.

Table 6.2 – Boundary conditions used for wave propagation with relaxation zones.

Field	Inlet\Outlet	Bottom	Top
U	waveVelocity	slip	pressureInletOutletVelocity
p_rgh	fixedFluxPressure	fixedFluxPressure;	totalPressure
alpha	waveAlpha	zeroGradient	inletOutlet
k	zeroGradient	zeroGradient	zeroGradient
omega	zeroGradient	zeroGradient	zeroGradient
nut	calculated	calculated	calculated

6.2 Error definition

6.2.1 Amplitude error

As for the wave propagation in a periodic domain (Sec. 5), the accuracy of the numerical simulations is evaluated analyzing the damping in time of the wave elevation. The wave elevation is measured at the center of the pure CFD domain. For a given time t a discrete Fourier transform is applied to the sampled surface elevation between $t_0 = t - 2.5T$ and $t_1 = t + 2.5T$. The resulting first-harmonic amplitude is noted $A_1^{CFD}(t)$. This first-harmonic amplitude of the surface elevation is compared to the analytic first-harmonic amplitude obtained with *CN-Stream* and noted A_1^{SF}. Then, the numerical error used in this section is defined as:

$$\mathscr{E}(t) = \frac{A_1^{CFD}(t) - A_1^{SF}(t)}{A_1^{SF}(t)} \tag{6.2}$$

6.2.2 Phase shift

The phase is evaluated using the same sliding-window discrete Fourier transform. The phase of the first-harmonic error at the center of the domain is noted ϕ^{CFD} when the reference analytical phase is noted ϕ^{SF} and obtained using *CN-Stream*.

6.2.3 Wave period

For the wave period, another method is used. A period T^{CFD} at a given instant t is defined as the average between two successive zero crossing in the same direction over a window of 5 periods between $t_0 = t - 2.5T$ and $t_1 = t + 2.5T$. T^{SF} is the analytical period obtained with *CN-Stream*.

6.3 Numerical setup

The reference numerical set-up (Sec. 3.6) is used. The investigation regards the temporal discretization, the mathematical models (NS or RANS) and the moving frame velocity. Table 6.3 synthesizes the tested configurations. Only the backward (`BCK`) and the Crank-Nicolson 0.95 (`CN95`) time schemes are tested. With RANSE resolution, the turbulent models k-ω-SST (`kOmegaSST`) and fs-k-ω-SST (`fsKOmegaSST`) are tested.

Table 6.3 – Numerical parameters studied with regular wave propagation with relaxation zones.

Numerical parameter	Tested values
`ddtScheme {default}`	`CranckNicolson 0.95` `backward`
`simulationType` (Model)	`laminar` (NS) `RAS kOmegaSST` (RANS) `RAS fsKOmegaSST` (RANS)
`fwdVelocity` (Frame velocity)	$(0, 0, 0)$ $(-\lambda/T, 0, 0)$
`deltaT`	$\left\{ \frac{T}{25 \times 2^k} \mid k \in [\|1, 4\|] \right\}$

6.4 Meshes

The meshes used in this study are the non-uniform meshes presented in Section 5.3 extended along the x-direction. Figure 6.3 shows a view of mesh `NU32x128` and Table 6.4 synthesizes the studied mesh configurations.

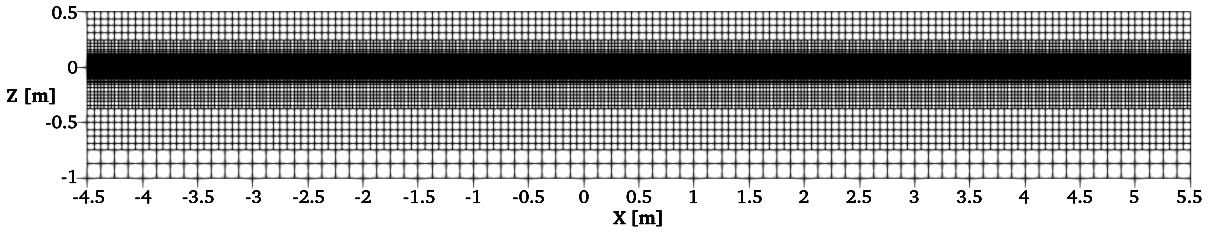


Figure 6.3 – non-uniform stretched mesh `NU32x128` used for wave propagation with relaxation zones.

Table 6.4 – Studied configuration of meshes for the regular wave propagation with relaxation zones.

Wave heights	$T/\Delta t$	**Mesh type**	**Mesh label**	$\lambda/\Delta x$	$\lambda/\Delta z$	$\mathrm{AR_{FS}}$
$H = 0.1$ m $H = 0.05$ m $H = 0.01$ m	{25, 50, 100, 100, 200, 400}	Non-uniform regular	NU32x32	32	32	1
			NU64x64	64	64	
			NU128x128	128	128	
			NU256x256	256	256	
		Non-uniform stretched	NU32x64	32	64	2
			NU32x128		128	4
			NU32x256		256	8
			NU32x512		512	16
			NU64x128	64	128	2
			NU64x256		256	4
			NU64x512		512	8
			NU128x256	128	256	2
			NU128x512		512	4
			NU256x512	256	512	2

6.5 Simulation in fixed frame with Navier-Stokes model

The first analysis is carried out using the Navier-stokes (NS) model, i.e. direct use of the Navier-Stokes equations without any turbulence modeling, and the domain configuration `RZi1o2`. The time histories are discussed first, then the phenomenon of vortices generation due to the relaxation zone and finally the error convergence and the influence of the free-surface cell aspect ratio ($\mathrm{AR_{FS}}$) are analyzed.

6.5.1 Time histories

Figure 6.4 shows the time history of the surface elevation first-harmonic amplitude error, the phase shift and the wave period error for two mesh resolutions. The solid and dotted lines correspond to simulations with `CN95` and `BCK`, respectively. The colors represent the time steps Δt.

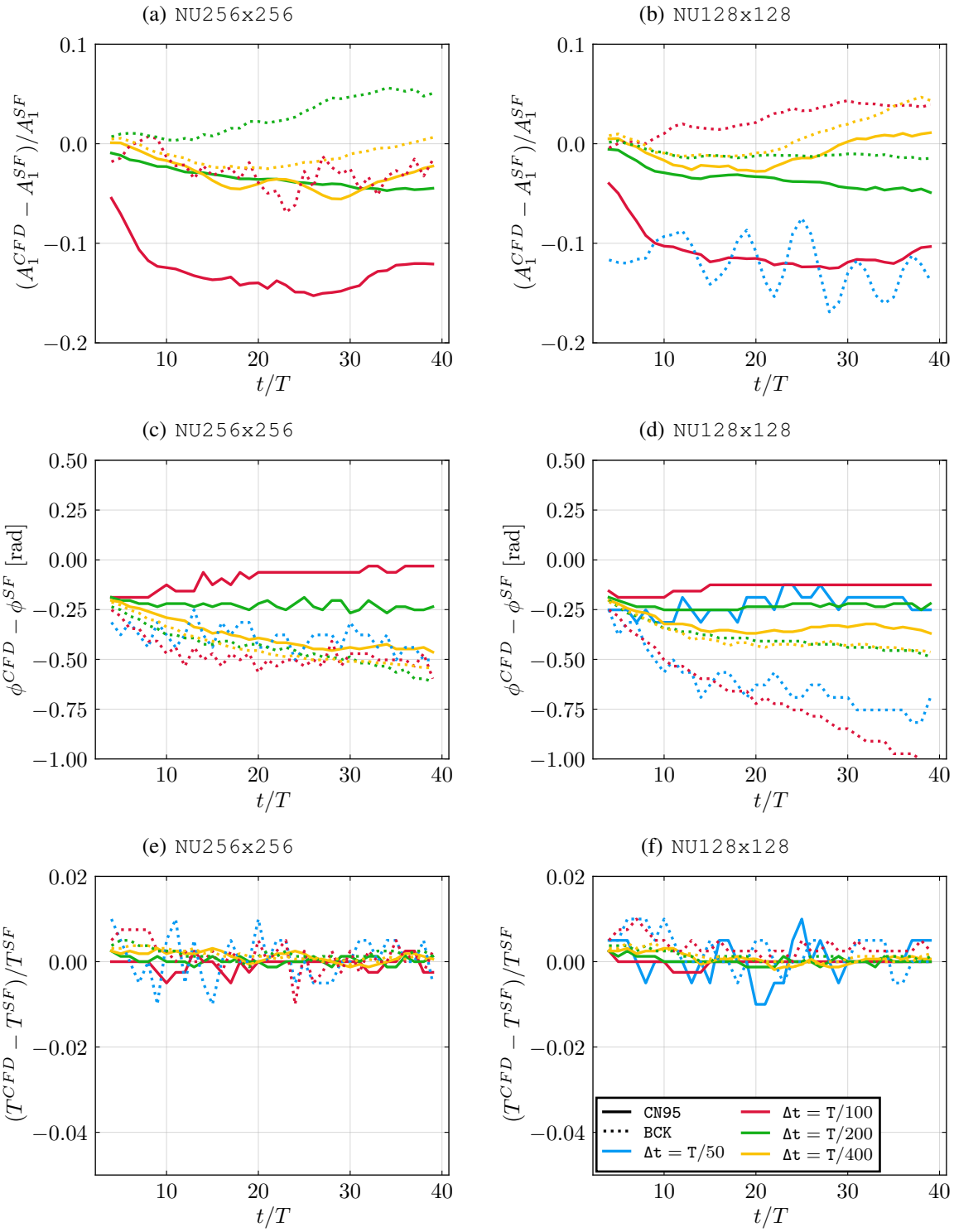


Figure 6.4 – Time histories of the surface elevation first-harmonic amplitude error, phase shift and wave periods for various time steps Δt with backward and Crank-Nicolson 0.95; regular wave; `H01`; `NU256x256` and `NU128x128`; Navier-Stokes model; domain `RZi1o2`.

First, the time histories of the amplitude error in Fig. 6.4 show that the evolution of the error is non constant. With some discretizations, the amplitude increases whereas with some others it decreases. For all the curves is that even if the amplitude decreases during the first periods, it tends to progressively increase after a certain time. Several causes could be involved in this phenomenon. The first one is the energy non-conservative resolution highlighted in Section 5. An other possible cause could be the presence of disturbance caused by the relaxation zones (this point is discussed in the following). Finally, both time schemes convergence is globally observed with a smaller damping of the wave elevation using `BCK` an some larger overshoots.

Then, the phase shift plotted in (c) and (d) reveals two things. The first one is a phase shift about 0.25 rad (about 4%) during the first periods. This shift is probably due to the initialization using a non physical velocity profile as discussed in Sec. 5. Besides, as for the amplitude error, the discretization influences significantly the evolution of the phase shift in manners that are not easily explainable. The only identifiable trend is that `BCK` causes larger phase shift than `CN95`.

Finally, the last two figures (e) and (f) show the time history of the wave period error. Unlike the wave amplitude error and the phase shift, the period error stays between -1% and 1% regardless of the discretization.

6.5.2 Generation of vortices in the air

In this section, investigation is done on the velocity fields in order to evaluate the impact of the relaxation zones. The studied wave is the steepest one $H = 0.1$ m; the mesh used is `NU256x256` and the time step of the simulation is $\Delta t = T/400$. In the following, the mainly observed phenomenon are noticeable using both `CN95` and `BCK`, so only the **BCK** scheme is investigated. Figure 6.5 shows the map of the horizontal velocity field after 10 periods and Figure 6.6 shows the norm of the vorticity. As the total size of the domain in 10λ, at $t = 10T$ the waves imposed at the initialization are supposed to be out of the domain so all the fields are depending from the inlet and outlet boundaries, and the relaxation zones.

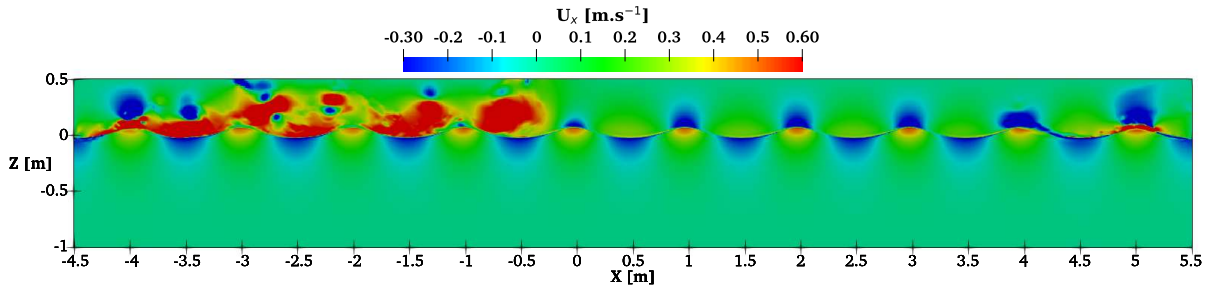


Figure 6.5 – Regular wave horizontal velocity field with backward time scheme at $t = 10T$; `NU256x256`; $\Delta t = T/400$; Navier-Stokes model; domain `RZi1o2`.

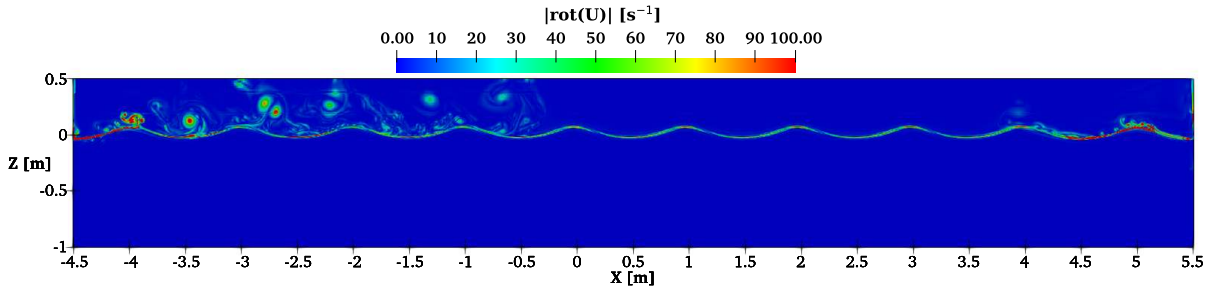


Figure 6.6 – Regular wave vorticity magnitude field with backward time scheme at $t = 10T$; `NU256x256`; $\Delta t = T/400$; Navier-Stokes model; domain `RZi1o2`.

Some disturbances in the air are observed in Fig. 6.5 and Fig. 6.6 between $x = -4.5$ m and $x \approx 0$ m. In this part of the domain vortices are visible and comparing to what has been observed in Sec. 5.4, the horizontal velocity reaches high values. Looking at Fig. 6.6, the source of the vortices seems to be the transition zone at the free surface in the inlet relaxation zone. In the remaining part of the pure CFD domain between $x \approx 0$ m and $x \approx 3.5$ m, the steady-state-like velocity field is progressively established and is consistent with the expected one (see Fig. 5.5). In the outlet relaxation zone, the air field is once again disturbed but with smaller vorticity than in the inlet zone.

This behavior is explained by the way the relaxation zone works. As detailed in Sec. 2.3.1, the relaxation imposes progressively the targeted values to the field, however, the imposed velocity is a null vector in the air. This null velocity is far from the one obtained with Navier-Stokes equations, consequently some large numerical corrections are applied by the solver in the relaxation zones. At the inlet side, the null velocity in air and the non-null wave velocity in water cause Kelvin-Helmholtz instabilities. These instabilities are then propagated in the domain. The vortices are progressively damped. However, in absence of turbulence model

and due to the lower Reynolds number in the air than in the water, the damping of these vortices remains small. After a few periods, the disturbed air is all along the domain. This phenomenon explains the noticed changes for the amplitude and phases after a certain period of time. First, the first-harmonic wave amplitude tends to decrease and the air velocity fields is converged. Then, the first-harmonic wave amplitude progressively increases. The hypothesis that should be explored is that the disturbances in the air phase cause an unsteady interaction with waves elevation when the vortices reach the measuring point.

6.5.3 Optimal free-surface cell aspect ratio and order of convergence

Despite the disturbance caused by the inlet relaxation zone, a convergence analysis is carried out over the three tested waves within the discretization range. Figure 6.7 shows the convergence of the first-harmonic amplitude error at fixed Courant numbers. As for the periodic wave study, two time-step/cell-size ratios are analyzed. Results with $\frac{\Delta t}{\Delta x} = \frac{128}{400}$ are shown in (a), (c), (e) and results with $\frac{\Delta t}{\Delta x} = \frac{128}{200}$ are shown in (b), (d), (f). Because of the reference velocity (Eq. 6.1) used for the Courant number calculation, the selected $\mathrm{Cfl_x}$ are different from one wave height to another. Solid and dotted lines represent Crank-Nicolson 0.95 and backward time schemes, respectively. Each color is associated to a free-surface cell aspect ratio $\mathrm{AR_{FS}}$ and each marker style corresponds to a cell size Δx. The gray dashed lines represent the first-order and second-order slopes.

Similar to the periodic waves, changes of $\mathrm{AR_{FS}}$ induce a modification of the accuracy. For a given wave height, some $\mathrm{AR_{FS}}$ with an optimal accuracy can be identified. Table 6.5 synthesizes these optimum values within the tested range. The results are close to the ones obtained with periodic waves (Tab. 5.6). Within the restricted range of spatial discretization only one point is plotted with $\mathrm{AR_{FS}} = 16$. It does not allow to conclude about the accuracy of this aspect ratio. Nevertheless, in accordance with the periodic wave analyses, for `H001` the value 16 is still indicated.

For a large part of the plots, the slopes are between first and second-order. A clear second-order behavior is hard to identify because of the previously identified perturbed air field. The damping of the perturbations in air decreases as the refinement increases and it paradoxically reduces the accuracy of the simulations.

Finally, even if for the finest spatial resolution $\Delta x = \lambda/256$ `NU256x256`, the backward scheme is more sensible than the Crank-Nicolson 0.95 to the perturbations in air, `BCK` is globally more accurate than the `CN95`.

Table 6.5 – Optimal aspect ratio for various wave steepness with relaxation zone and Navier-Stokes model.

Wave	Steepness	Optimal aspect ratio
H001	1%	8 or (16)
H005	5%	4
H01	10%	2

These optimal aspect ratios follow the same conclusion as the one made in with the periodic waves analyses. So, the general recommendation for the free-surface cell aspect ratio simulating regular waves with relaxation zones is also:

$$\frac{\lambda}{5H} \leq \mathrm{AR_{FS}}^{Opti} \leq \frac{\lambda}{10H} \tag{5.3}$$

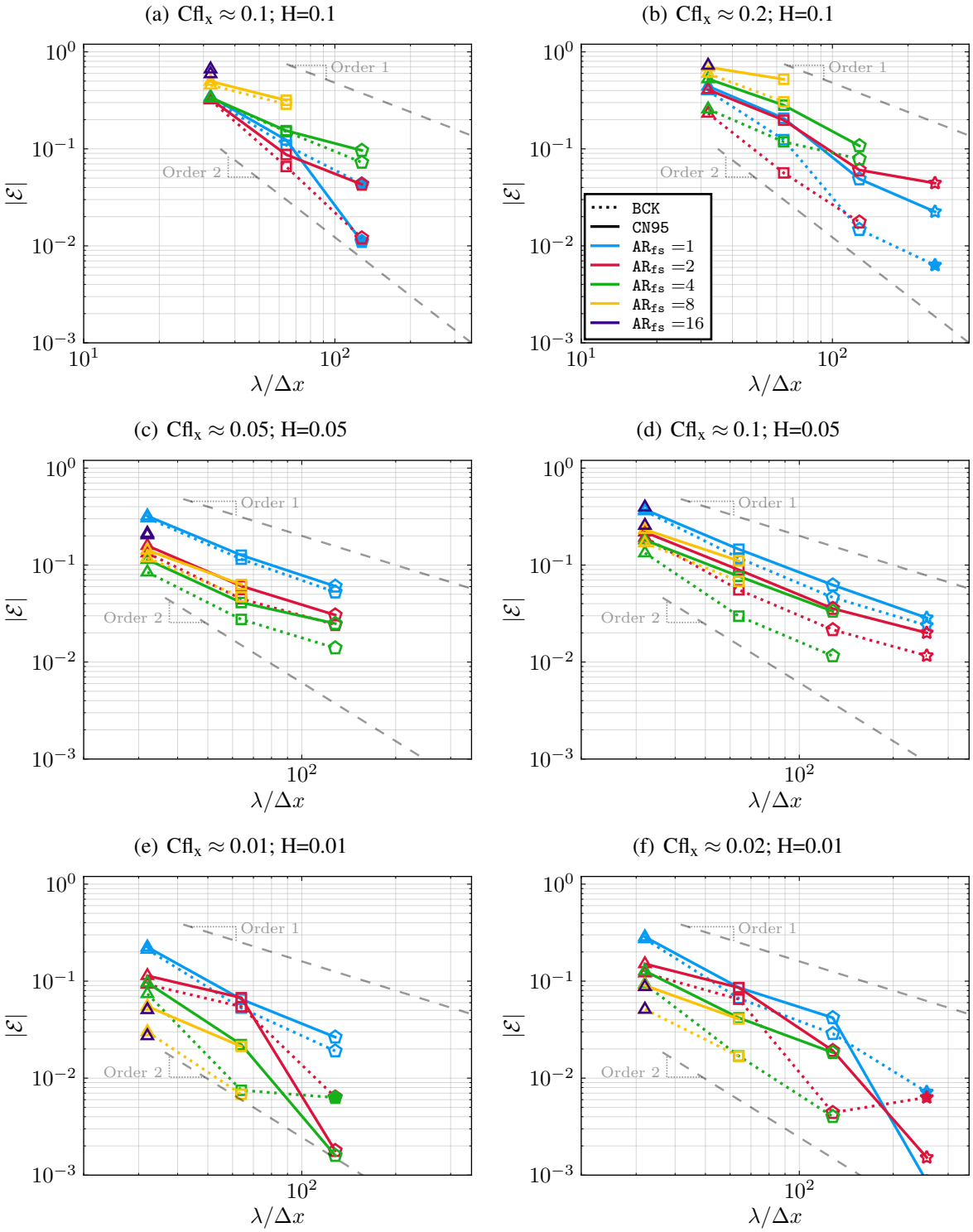


Figure 6.7 – Surface elevation first-harmonic amplitude error at $t = 37.5T$ depending on cell aspect ratio at the free surface (AR_{FS}) for two Courant number Cfl_x and for backward and Crank-Nicolson 0.95; Navier-Stokes model, domain `RZi1o2`.

6.6 Simulation in fixed frame with RANS model

At this stage, the accuracy of *foamStar* simulating nonlinear regular waves with a backward time scheme and the `MULES` algorithm has been shown. However, the NS model is insufficient for naval applications because of the very high Reynolds number at stake, implying very thin boundary layers and the turbulent flow around the hulls. So, in this section, using `BCK`, the results with the two RANS models are compared to the NS one. A discussion about the undesirable vortices management is done, firstly considering the turbulent models and then the domain configurations. Besides, the surface elevation errors are examined.

6.6.1 Influence of the turbulent model on the air vortices

a) OpenFOAM k-ω-SST

Figure 6.8 shows a map of the norm of the vorticity field after ten periods using RANS with the k-ω-SST turbulence model from OpenFOAM. In this figure no vortices are noticeable. The reason is the turbulence production by the model where the vortices are created. The side effect is that this turbulent production damps the wave such that in the pure CFD zone the wave elevation is 70% damped after only ten periods, see Figure 6.9

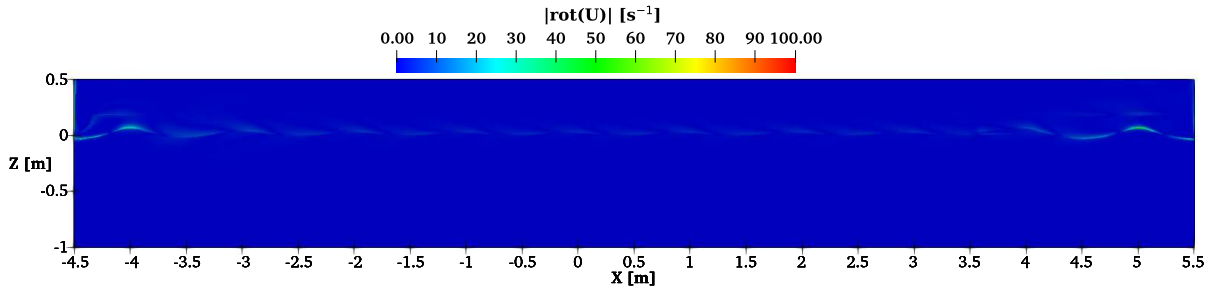


Figure 6.8 – Regular wave vorticity magnitude field with backward time scheme at $t = 10T$; `NU256x256`; $\Delta t = T/400$, RANSE $k-\omega$ SST; domain `RZi1o2`.

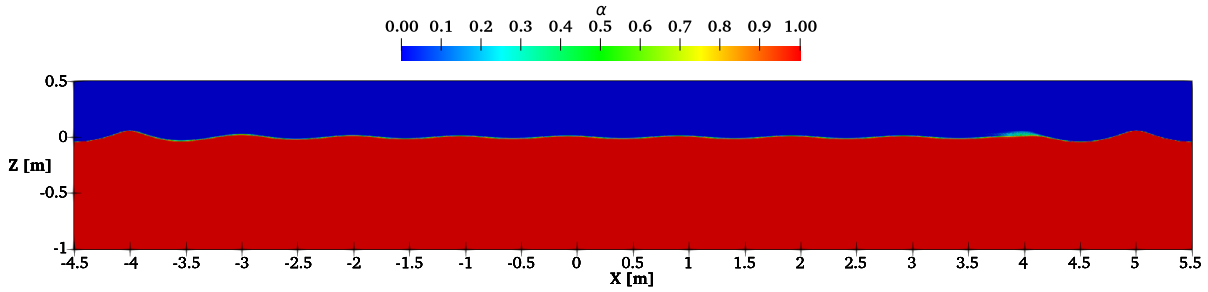


Figure 6.9 – Regular wave volume-fraction field with backward time scheme at $t = 10T$; `NU256x256`; $\Delta t = T/400$, RANS k-ω-SST; domain `RZi1o2`.

b) *foamStar* fs-k-ω-SST

As detailed in Section 2.1.3 an alternative to the OpenFOAM k-ω-SST model is the fs-k-ω-SST implemented in *foamStar*. Figure 6.10 shows the map of the norm of the vorticity field after 10 periods using the fs-k-ω-SST RANS turbulence model. Figure 6.11 shows the corresponding horizontal velocity field. The magnitude of the vorticity is smaller than with NS simulations (Fig. 6.6). However, vortices are still present and disturb the horizontal velocity field. Compared to the OpenFOAM k-ω-SST turbulence model, the fs-k-ω-SST model limits the turbulence production and therefore the damping.

As a conclusion, at this point none of the three options is satisfactory. First, by construction the NS model does not model correctly the turbulent part of the flow and the large spurious vortices created in the inlet zone degrade the wave propagation quality. Then, a standard turbulence model damps those vortices but at the cost of damping also the gravity waves , since the free-surface boundary layer is not correctly accounted for in that model. Finally, the free-surface turbulent model, which is the most physically correct, does not damp much the spuriously created inlet vortices.

Remark: *Additionnal works have shown that applying the relaxation update of the velocity field after the PISO loop, as it is done in the current foamStar version, significantly reduces the vorticies generation. This effect has been noticed after the present study so this solution is not explored in this document.*

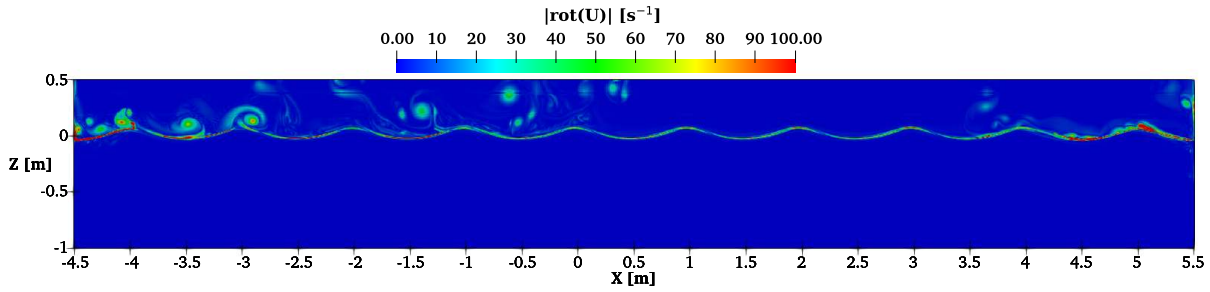


Figure 6.10 – Regular wave vorticity magnitude field with backward time scheme at $t = 10T$; `NU256x256`; $\Delta t = T/400$, RANS fs-k-ω-SST; domain `RZi1o2`

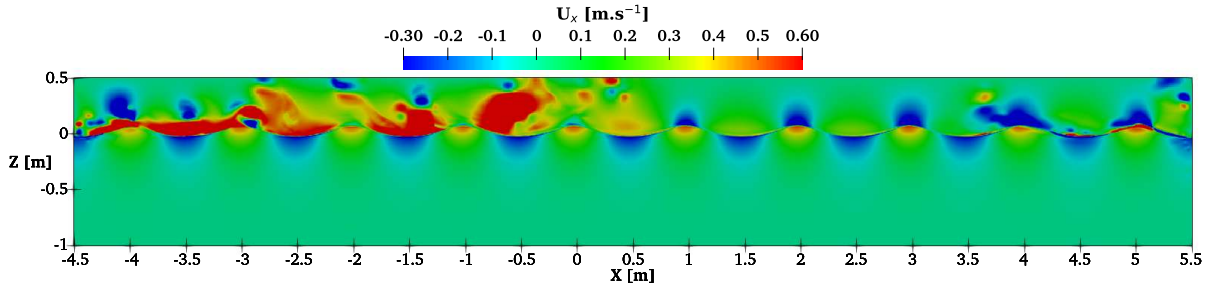


Figure 6.11 – Regular wave vorticity magnitude field with backward time scheme at $t = 10T$; `NU256x256`; $\Delta t = T/400$, RANS fs-k-ω-SST; domain `RZi1o2`

6.6.2 Removing the inlet relaxation zone

In order to remove the perturbations induced by the inlet relaxation zone, the domain `RZo3` is tested. With `RZo3` there is no inlet relaxation zone and the wave is only generated by the inlet boundary condition. The air velocity field imposed at the boundary is still null but it is expected that the Kelvin-Helmholtz instabilities and thus the vortices production would be limited. Since shearing between waves and the incorrect zero-velocity condition in air is only acting on one boundary point rather than on a contact surface.

Figure 6.12 and Figure 6.13 show the norm of the vorticity and the horizontal wave field, respectively, after ten periods. The simulations are done using RANS fs-k-ω-SST turbulent model. Fig. 6.12 shows the absence of vortices production at the inlet. Some are still produced at the outlet. In Fig. 6.13 the steady-state-like horizontal velocity field is established from $x = -4.5$ m to $x \approx 3$ m.

Removing the inlet relaxation zone prevents the production of non-physical vortices and then the turbulence production at the inlet of the domain. So, for naval applications with forward

speed, in order to avoid both the perturbation of the free surface and an additional drag in the air, using only an outlet zone could be convenient. The relaxation zones are used to damp the diffracted fields, so with a body, the inlet relaxation zone is designed to avoid reflections of the diffracted fields on the inlet boundary. For a floating body without forward speed this diffracted field could be non-negligible. However, thanks to the target naval applications, in the following the selected configuration is `RZo3` with fs-k-ω-SST turbulent model.

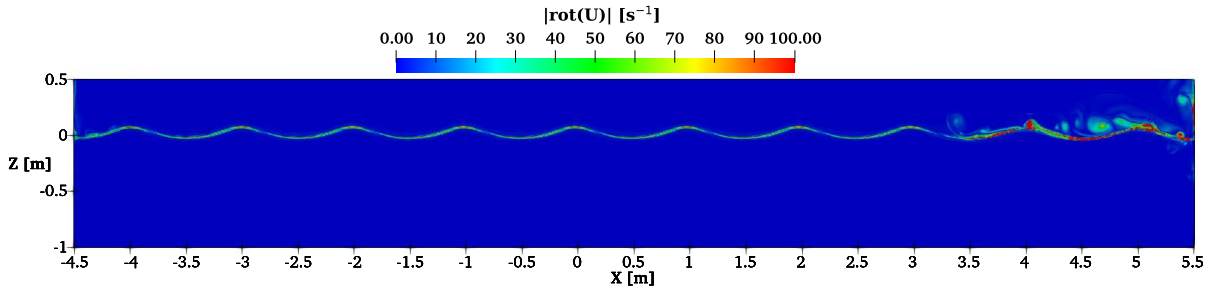


Figure 6.12 – Regular wave vorticity magnitude field with backward time scheme at $t = 10T$; `NU256x256`; $\Delta t = T/400$, RANS fs-k-ω-SST; domain `RZo3` (no inlet relaxation zone).

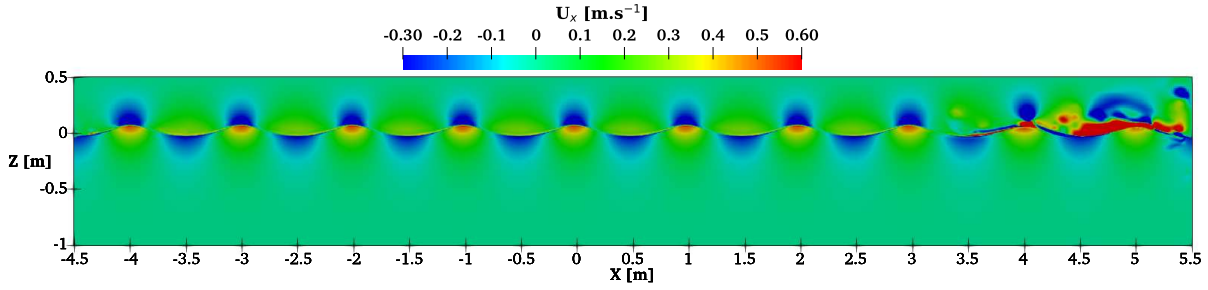


Figure 6.13 – Regular wave vorticity magnitude field with backward time scheme at $t = 10T$; `NU256x256`; $\Delta t = T/400$, RANS fs-k-ω-SST, without inlet relaxation zone; domain `RZo3`.

6.7 Simulation in moving frame with RANS model

The next step toward naval simulations is to test the influence of moving frames with imposed forward speed. For the computation of added resistance and ship motions, the waves are simulated in a moving frame. This section presents a qualitative error comparison between fixed and moving-frame simulations.

Firstly, the map of the vorticity norm after ten periods shown in Figure 6.14 reveals an higher vortices productions with forward speed. Even if no inlet relaxation zone is used,

votrices are distributed all along the domain. The boundary condition at the inlet could be a cause of this. An other reason could be the higher Courant number caused by the forward velocity and inducing some spurious vortices. Finally, the vortices generated in the outlet zone could be reversely propagated because of some reflections at outlet boundary. In such a case, potential reflections at the inlet should also be discussed.

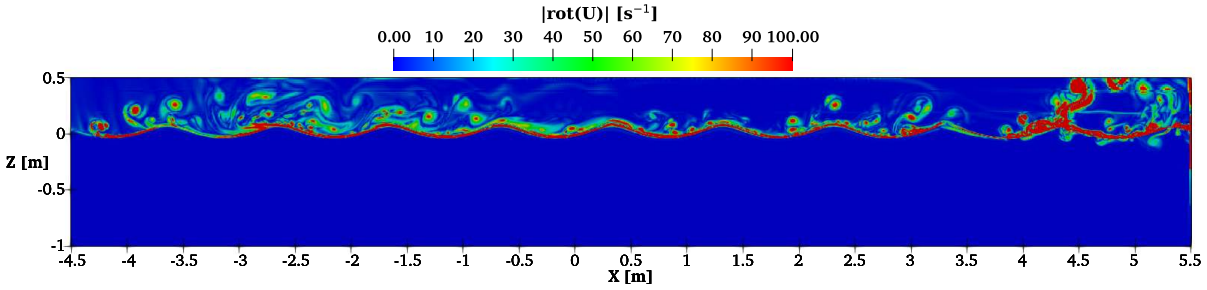


Figure 6.14 – Regular wave vorticity magnitude field with backward time scheme at $t = 20Te$; `NU256x256`; $\Delta t = T/400$, RANS fs-k-ω-SST; domain `RZo3`.

Figure 6.15, Fig. 6.16 and Fig. 6.17 are `iso-time-step views` of the surface elevation first-harmonic amplitude error from simulations of nonlinear regular waves with a steepness of 10%, 5% and 1%, respectively. In Fig. 6.15 and Fig. 6.16, the free surface cell aspect ratio are $\mathrm{AR_{FS}} = 2$ and $\mathrm{AR_{FS}} = 4$, respectively. They correspond to the recommendation from Tab. 6.5. For `H001`, because of the lack of spatial discretization with $\mathrm{AR_{FS}} \geq 8$ is not convenient for the accuracy visualization. So, though it is not in the optimal range previously defined, $\mathrm{AR_{FS}} = 4$ is used in Fig. 6.17. All the simulations are done using RANS fs-k-ω-SST models and the domain configuration `RZo3`. The colors represent the time resolution with respect to the encountered wave period T_e. In these figures, for a given wave, `V0` denotes the fixed frame with a null velocity $V_0 = 0$ m.s^{-1} and `V1` denotes the frame moving toward negative x-direction at the phase velocity $V_1 = \dfrac{\lambda}{T}$. The Courant numbers are adapted according to the forward speed with the following definition of the reference velocity U_{ref}

$$\mathrm{Cfl_x} = \frac{(U_{\mathrm{ref}} + V_i)\Delta t}{\Delta x}. \tag{6.3}$$

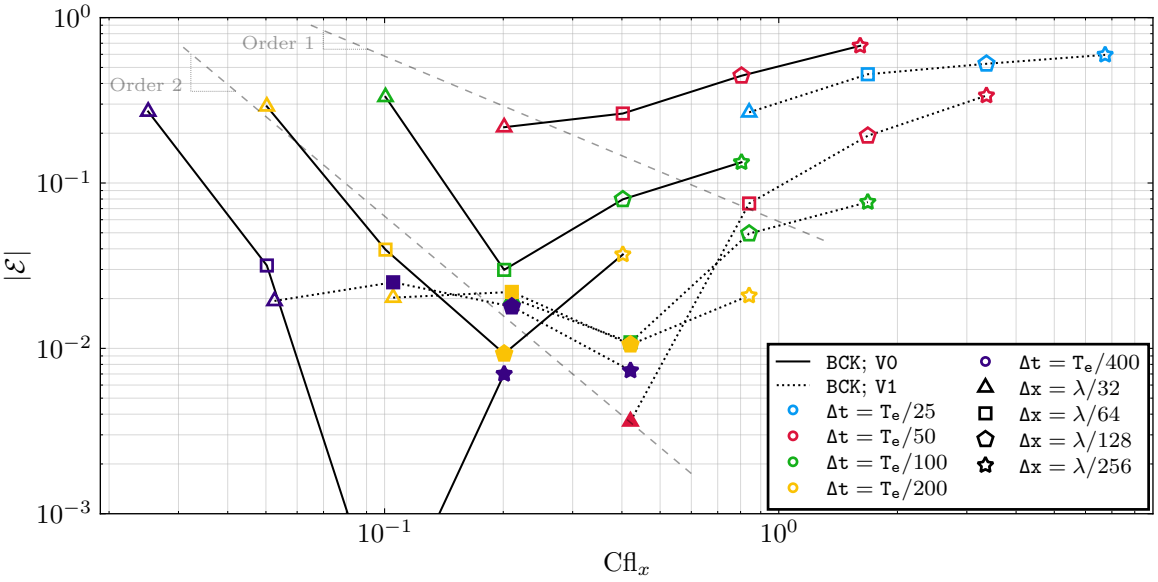


Figure 6.15 – *Iso-time-step view* of the surface elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number $\mathrm{Cfl_x}$ for various time steps Δt with fixed and moving frame; RANS fs-k-ω-SST; `H01`; $\mathrm{AR_{FS}}$=2; backward.

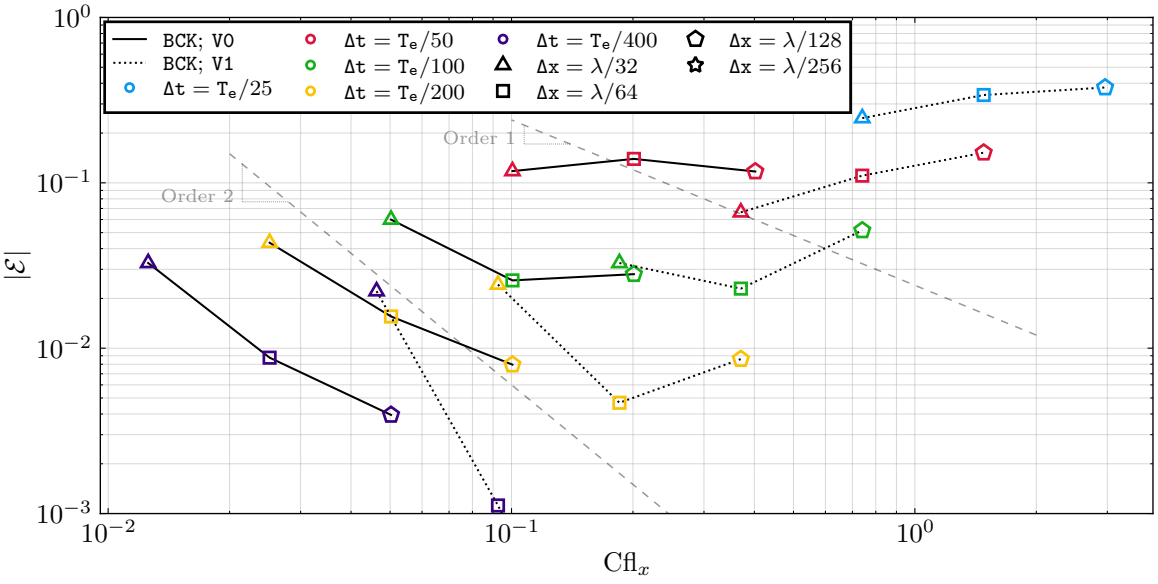


Figure 6.16 – *Iso-time-step view* of the surface elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number $\mathrm{Cfl_x}$ for various time steps Δt with fixed and moving frame; RANS fs-k-ω-SST; `H005`; $\mathrm{AR_{FS}}$=4; backward.

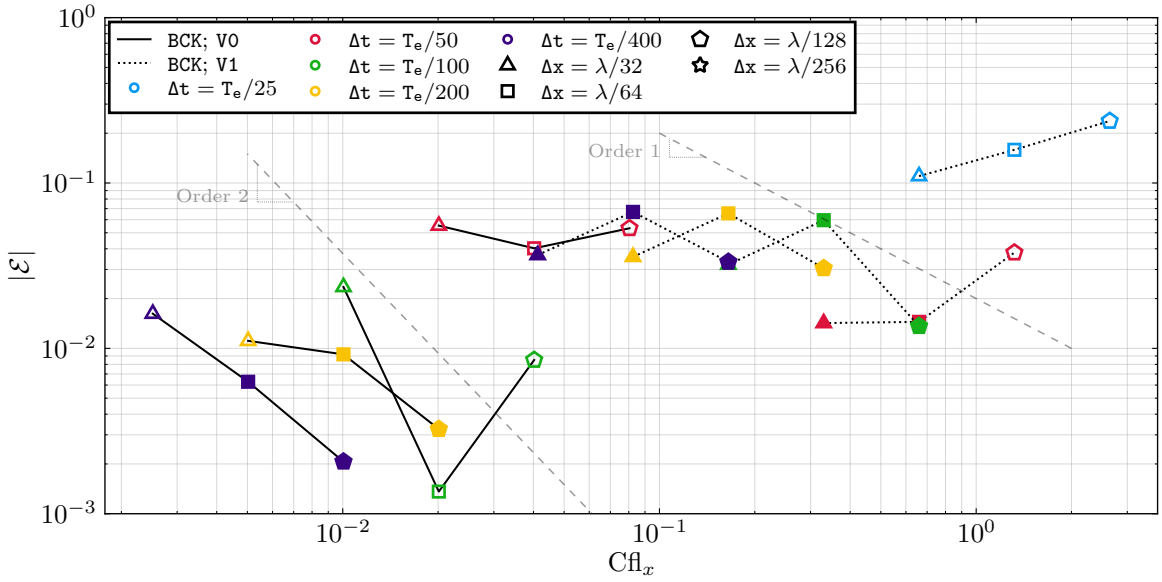


Figure 6.17 – *Iso-time-step view* of the surface elevation first-harmonic amplitude error at $t = 37.5T$ depending on Courant number $\mathrm{Cfl_x}$ for various time steps Δt with fixed and moving frame; RANS fs-k-ω-SST; `H001`; $\mathrm{AR_{FS}}$=4; backward.

In Fig. 6.15 presenting the wave `H01`, an increase of the error is noticeable when the cell size decreases for $\Delta t \geq T_e/100$, with both the fixed and moving frame. With the moving frame the accuracy of $\Delta x = \lambda/32$ is greater than with a fixed frame. Nevertheless, with $\Delta t \leq T_e/200$ the error stagnates regardless of the time and space resolutions, for this moving frame. An hypothesis that might explain this phenomenon is that the gain of energy identified in Sec. 5.4 could be amplified by the increase velocity magnitude due to the forward speed. Then, this gain of energy due to non-conservation would counterbalance the numerical damping. The saturation of the error might also be caused by an increase of the Courant number due to the forward speed. The vortices production amplified with the forward speed is also likely to be involved. Exploring these tracks is left to future work.

Overall, the error with the forward speed `V1` is smaller than the error with fixed frame `V0` for more than 80 % of the tests. However, as all the identified effects are not well quantified, no conclusions about the accuracy gain can be done.

With `H005` (Fig. 6.16) as with the other waves, for a given time step the error with moving frame simulation does not decrease when the cell size decreases. However, the time step influences are closer to the expectations, the smaller the time step the smaller the erro is.

Finally, with `H001` (Fig. 6.17) the non decreasing error regardless of the time and space resolution can be noticed and with this wave, for the finest resolutions the error with forward

speed is larger than the error with fixed frame. This different behavior might be due to the choice of AR_{FS} that is not assessed as an optimal one.

6.8 Efficiency

The comparison of the efficiency is the final step of the analyses of the wave generation in a domain using relaxation zones with RANS model. Figure 6.18, Fig. 6.19 and Fig. 6.20 show the *iso-cell-size views* of the error but using the CPU time for the abscissa rather than Cfl_x. The figures concern the waves `H01`, `H005` and `H001`, respectively. The free-surface cell aspect ratios are $AR_{FS} = 2$, $AR_{FS} = 4$ and $AR_{FS} = 4$, respectively. The scheme used is still the fs-k-ω-SST.

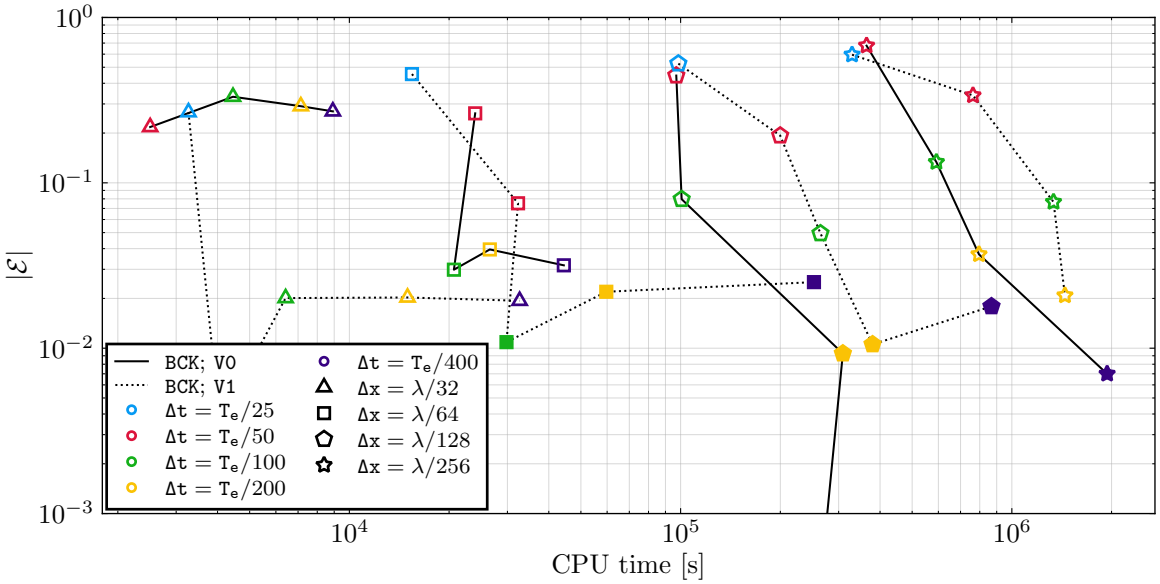


Figure 6.18 – *Iso-cell-size view* of the surface elevation first-harmonic amplitude error at $t = 37.5T$ and CPU time for various cell sizes Δx with fixed and moving frame; RANS fs-k-ω-SST; `H01`; AR_{FS}=2; backward.

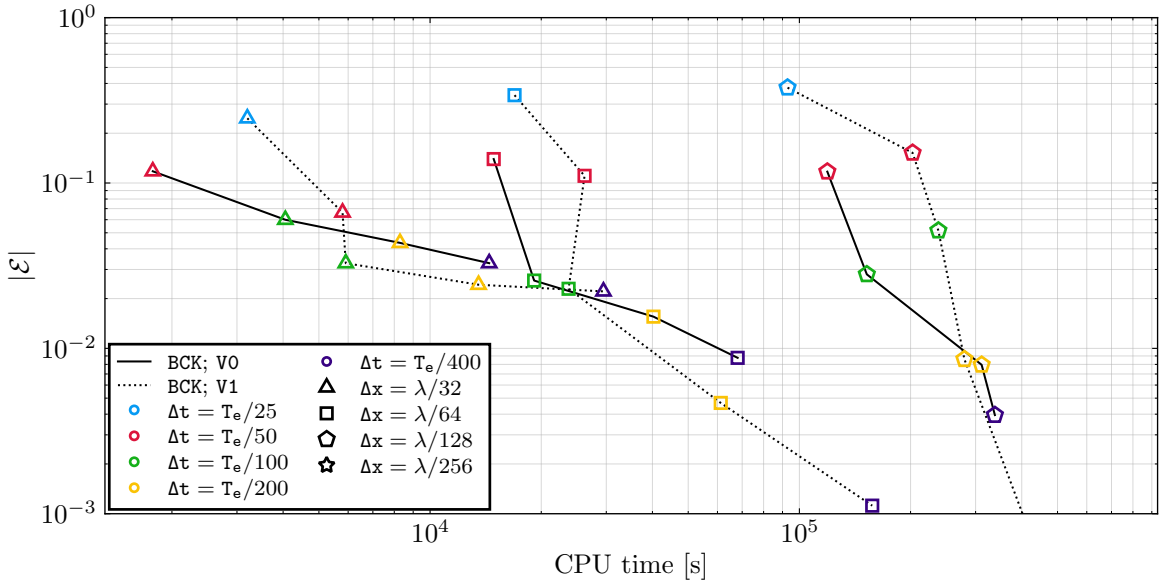


Figure 6.19 – *Iso-cell-size view* of the surface elevation first-harmonic amplitude error at $t = 37.5T$ and CPU time for various cell sizes Δx with fixed and moving frame; RANS fs-k-ω-SST; `H005`; AR_{FS}=4; backward.

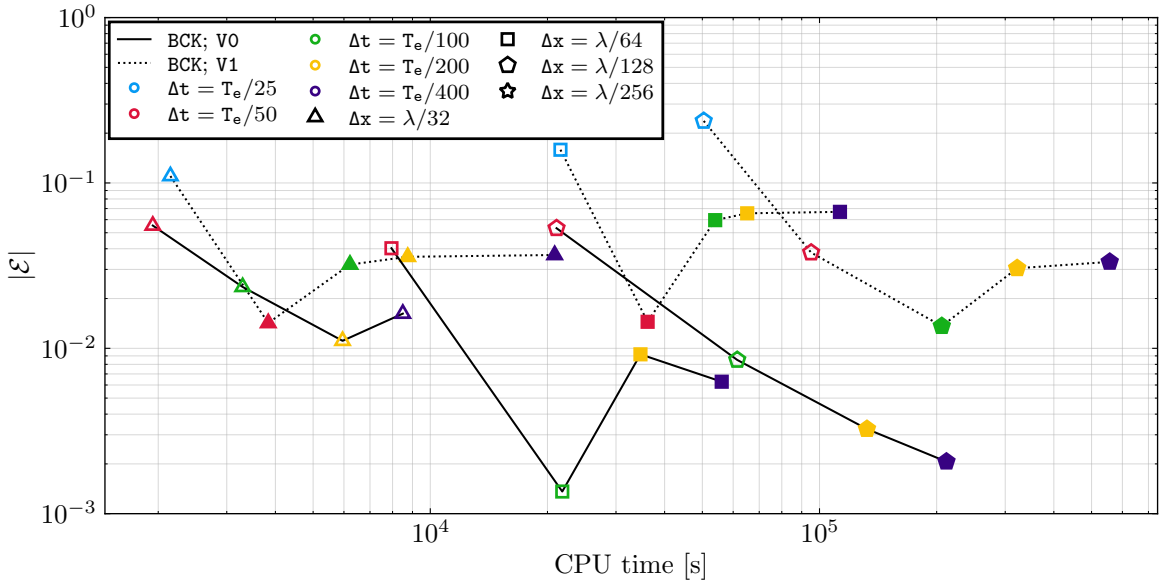


Figure 6.20 – *Iso-cell-size view* of the surface elevation first-harmonic amplitude error at $t = 37.5T$ and CPU time for various cell sizes Δx with fixed and moving frame; RANS fs-k-ω-SST; `H001`; AR_{FS}=8; backward.

The errors are the same as the ones shown in Fig. 6.15, Fig. 6.16 and Fig. 6.17, so

the saturation of the error with `V1` is still noticeable. The objective here is to identify some guidelines that could be convenient for simulations with forward speeds between `V0` and `V1`. The apparent efficiency of $\Delta x = \lambda/32$ for simulations with `V1` cannot be taken as a reference as it is not the case with `V0`. Besides, with the current *foamStar* implementation, the increase of perturbations reducing the time step and the cell size implies that taking a too small refinement would not be cost effective. Consequently, the recommendation is to use $\Delta x^{Opti} \approx \lambda/64$ and $\Delta t^{Opti} \approx T_e/200$ with configurations and mesh similar to those used in this study. Note that, contrary to the other previous recommendations, that one should be taken with caution, and is likely to evolve with future studies.

6.9 Conclusion

For simulations of regular wave inside domains with relaxation zone it has been shown that the relaxation zone technique currently implemented in `foamStar` generates some spurious vortices in the air that can disturb the water field and affect the convergence. These vortices are caused by the targeted null air velocity inside the relaxation zones that is far from the converged Navier-Stokes physical solution in the pure CFD domain. These vortices are attenuated using RANSE with fs-k-ω-SST turbulent model and a convenient option for a simualtion with forward speed is to remove the inlet relaxation zones only using the boundary condition to generate the waves. This solution could be further be discussed evaluating its consequence on the reflections of diffracted waves on the inlet boundary. Besides, the influence of the free surface cell aspect ratio as been shown and the recommendations made with the periodic wave study are confirmed. It also has been shown that the use of moving frame leads to additional errors and non clear convergence within the tested range of discretization. The following time and space resolution recommendations have been made:

$$\Delta x^{Opti} \geq \frac{\lambda}{64} \tag{6.4a}$$

$$\Delta t^{Opti} \geq \frac{T_e}{200} \tag{6.4b}$$

$$5 \leq \frac{\lambda\, \Delta z^{Opti}}{H\, \Delta x^{Opti}} \leq 10 \tag{6.4c}$$

Finally, comparing to the work done in Kim (2021) and Kim et al. (2022a), some conclusions are quite different. For instance the influence of the choice of RANS or NS

model is less noticeable in Y.J, Kim's work. Then, the problem of vortices disturbing the wave propagation when inlet relaxation zones are used was not encountered to such an extent. A major difference between the presented work and Kim's work is the mesh refinement method. In this work both the longitudinal and vertical cell refinement become coarser close to the upper and lower boundaries and the refinement coefficient [3] is 2. In Kim's work, the horizontal cell size is constant and the vertical refinement is progressively changed using grading methods and coefficients between 1.01 to 1.1. This might be a one cause of the noted differences, so for future work the numerical set-up tested in this chapter should also be tested with Kim's meshes.

3. The ratio between the size of two neighboring cells.

DISCUSSION ON THE NUMERICAL STUDY

Over this part, a progressive analysis has been done from the Navier-Stokes simulation of the analytic mono-fluid Taylor-Green vortices to RANS-VOF simulations of nonlinear regular waves with relaxation zones and forward speed. The objectives of these studies were to assess the accuracy and the efficiency of *foamStar* with the new second-order backward MULES implementation, to identify the influence of the VOF, the turbulence model and the frame velocity on the accuracy and to define some best practices for naval simulations with regular waves.

First, the accuracy of the second-order backward scheme has been verified, first in mono-fluid domain and then with the multi-phase simulations of regular waves using the bakward MULES. Comparing it to the commonly used Crank-Nicolson 0.95, the backward scheme shows better accuracy.

Besides, the ability of *foamStar* to simulate nonlinear regular waves with RANS has been assessed. it has been especially shown that using a modified standard turbulence solver to account for the free-surface presence is highly beneficial. However, issues have been identified with the model used for the wave generation. Using relaxation zones, a targeted null velocity in air causes some spurious vortices in the air. Some other kinds of extrapolations have been proposed by Choi (2019) for irregular wave propagation. However, the latter methodss have not been tested with regular waves and some air solutions could be established more accurately (potential solution in the air, pre-solving of a 2D NS solution ...). Using the currently implemented relaxation zones, using only an outlet one with just enforcing the BCs at inlet has shown some improvements and would be convenient for naval simulation with forward speed.

Finally over all the studied cases, a common observation was the strong influence of the mesh structure on the accuracy. The influence of some mesh deformations have been measured and showed that with *foamStar*, some shapes of cells decrease a lot the accuracy when others are well managed. For mono-fluid cases using non-orthogonal corrections, the cell aspect ratio and the non-orthogonality cause only small disturbances whereas the skewness from tested `zigZag` deformation induces significant error. For mono-fluid cases, the skewness corrector has been tested and has shown a significant improvement of the accuracy. For the simulation of regular waves with VOF, only the aspect ratio close to the free-surface has been studied

revealing its significant influence on the accuracy of the wave propagation. Some optimal aspect ratios depending on the wave steepness have been identified.

PART III

Naval application

---

foamStar is already used by Bureau Veritas Marine & Offshore (BV-M&O) for some naval applications. However, the company supervising this Ph.D., Bureau Veritas Solutions Marine & Offshore (BVS-M&O), did not use foamStar in their industrial process at the start of the PhD. This part presents the work done in order to qualify and improve the accuracy of foamStar for some naval simulations and its ability to be used in an industrial context. The first chapter presents a comparison between foamStar and foamStar-SWENSE to evaluate added wave resistance and ship motions in head waves. The content of this chapter was generated during the first year of the PhD, as an initial assessment of the capability of the code. The last section present some results obtained at the very end of the PhD, after the work on the code and the learning from the dedicated systematic numerical study. The second chapter details the work done in order to ease the industrial use of foamStar.

CHAPTER 7

NAVAL SIMULATION IN HEAD WAVES

In this chapter the numerical analysis are conducted using the Korea Research Institute for Ships and Ocean Engineering (KRISO) Container Ship (KCS). Numerical results obtained with the *foamStar* and *foamStar-SWENSE* solvers are compared to experimental data in regular wave conditions with 3DoF motions and forward speed. A seakeeping simulation in a rough irregular sea state is conducted using *foamStar-SWENSE* in order to highlight the robustness of the solver.

7.1 Case definition

This section presents the cases studied all along this chapter. Both the geometry and the regular wave conditions come from the case 2.10 of the CFD Tokyo 2015 Workshop (Larsson et al., 2018).

7.1.1 Geometry and test conditions

The studied geometry is a scaled model (1:37.89) of the KCS. This ship is a modern container ship with a bulbous bow used in many numerical and experimental investigations. The main features of the KCS geometry at full and model scale are given in Table 7.1.

Table 7.1 – Main particulars of KCS geometry

Main particulars	**Full**	**Model**
Length (L_{PP}) [m]	230	6.0702
Breadth (B_{WL}) [m]	32.2	0.8498
Depth (D) [m]	19.0	0.5015
Draft (T) [m]	10.8	0.2850
Volume (Δ) [m^3]	52030	0.9571
Wetted surface (S_W) [m^2]	9539	6.6978
Horizontal buoyancy center (L_{CB}) [m]	111.5	2.944
Vertical center of gravity (K_G) [m]	14.32	0.378
Radius of Inertia $(\frac{K_{xx}}{B})$	0.40	0.40
$(\frac{K_{yy}}{L_{PP}}, \frac{K_{yy}}{L_{PP}})$	0.250	0.252

The forward speed condition is presented in Tab. 7.2 together with the associated non dimensional numbers computed in the experimental conditions.

Table 7.2 – Physical parameters for KCS experiments and numerical simulations

Ship speed (U_0) [m.s^{-1}]	**Fr**	**Re**
2.017	0.261	$1.074 \cdot 10^7$

For the following simulations, only 2 DoF are considered, see Tab. 7.3.

Table 7.3 – Degrees of freedom during KCS experiments and numerical simulations

Surge	**Sway**	**Heave**	**Roll**	**Pitch**	**Yaw**
Towing (no spring)	*Fixed*	*Free*	*Fixed*	*Free*	*Fixed*

The six sea conditions from case 2.10 of the Tokyo 2015 workshop are treated. One is a calm water condition and five are regular head waves. These conditions are detailed and labeled in Table 7.4 where notations are λ for the wavelength, T for the period, T_e for the encountered the wave period, H for wave height and $\varepsilon = H/\lambda$ for the steepness.

Table 7.4 – Regular Waves conditions for KCS model scale hull

Case	λ [**m**]	$\frac{\lambda}{\mathbf{L_{PP}}}$	H [m]	T [s]	T_e [s]	ε
C0	Calm water					
C1	3.949	0.651	0.062	1.591	0.878	0.016
C2	5.164	0.851	0.078	1.819	1.063	0.015
C3	6.979	1.150	0.123	2.115	1.311	0.018
C4	8.321	1.371	0.149	2.309	1.479	0.018
C5	11.840	1.951	0.196	2.754	1.873	0.017

The experimental data relevant to the studied cases were provided for the Tokyo 2015 Workshop. It was issued from the towing tank experiments realised by Van et al. (1998), Kim et al. (2001) and updated by Larsson et al. (2013).

7.1.2 Numerical Setup

Figure 7.1 illustrates the computational domain geometry and the boundaries used in the KCS study. The boundary conditions are specified in Table 7.5.

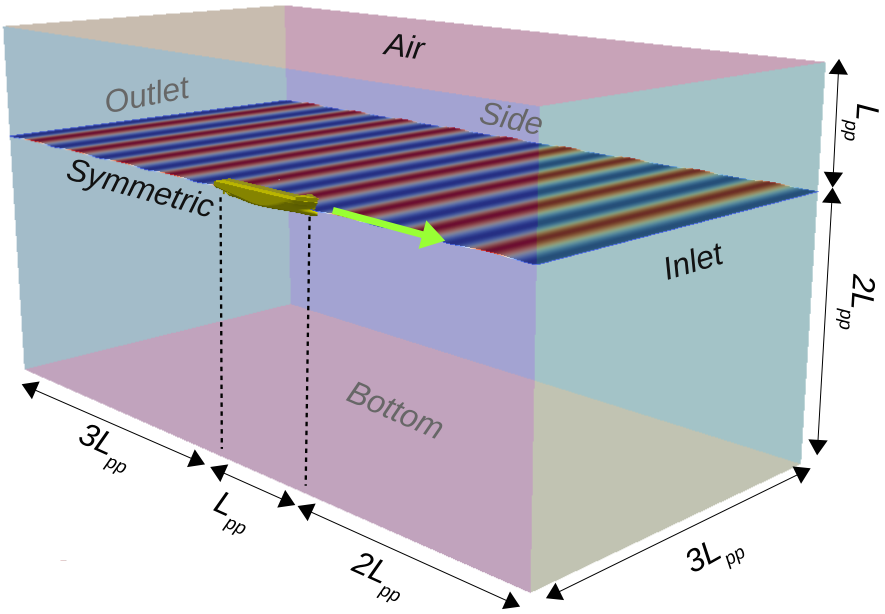


Figure 7.1 – Numerical domain for KCS simulations

Table 7.5 – Boundary conditions used for naval simulation in head waves.

Field	Inlet\Outlet	Bottom	Top
U	waveVelocity	slip	pressureInletOutletVelocity
alpha	waveAlpha	zeroGradient	inletOutlet
p_rgh	fixedFluxExtrapolatedPressure		totalPressure
k	zeroGradient		
omega	zeroGradient		
nut	calculated		

7.2 First assessment with regular head waves

This section presents a study carried out before the improvements of the numerical accuracy of *foamStar*. The following is extracted from the paper "Seakeeping in regular and irregular waves with forward speed using a two-phase functional coupling based SWENSE solver" presented during the 33rd Symposium on Naval Hydrodynamics in October 2020.

The objective is to qualify the ability of the solver to handle naval simulations of ship with forward speed in head regular and irregular waves. Both *foamStar* and *foamStar-SWENSE* are tested in this section. The SWENSE method tested in this section is the one exposed in (Li et al., 2021). This method has shown some efficiency gains by reducing the mesh refinement requirements and consequently the computational cost (Li et al., 2018b; Kim, 2021).

7.2.1 Settings

The configuration of the relaxation zones is shown in Figure 7.2. Relaxation zones are used at the inlet, outlet and on the side of the ship. The width of the relaxation zones is one L_{PP}. This configuration, used for an initial assessment, differs from the one prescribed in the previous chapters of this document.

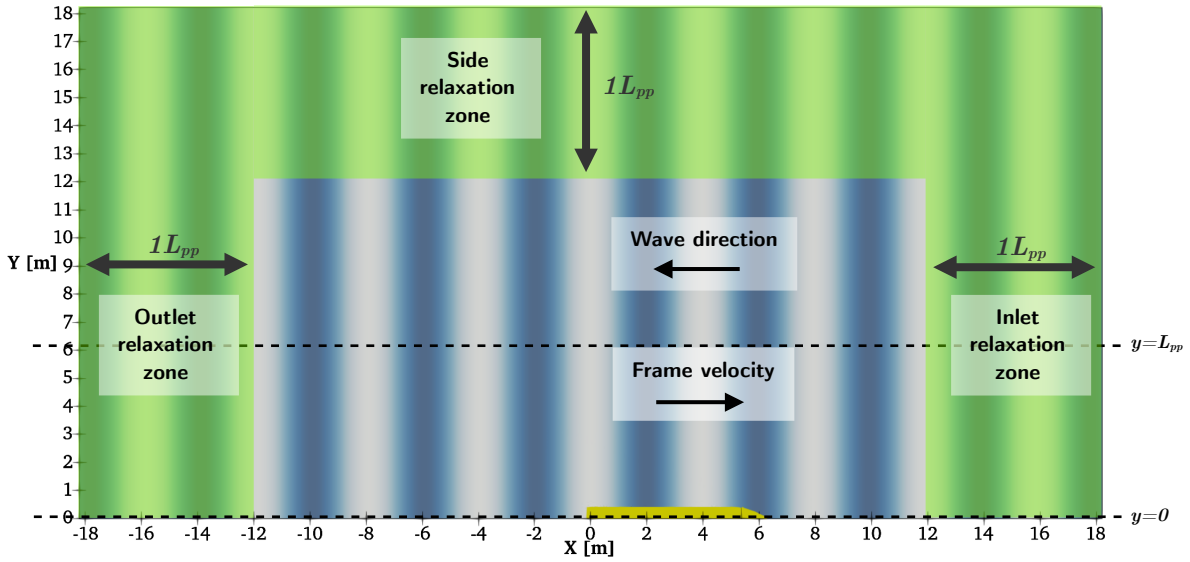


Figure 7.2 – Relaxation zone configuration for KCS simulations

Table 7.6 presents some numerical parameters selected for this study. When this study was conducted, the backward MULES scheme was not available and the Crank-Nicolson 0.95 scheme caused some simulations to fail. Consequently, for all the simulations presented in this section, only the Euler time scheme is used. Besides, the turbulent model used is not fs-k-ω-SST but k-ω-SST. The time steps are fixed to $\Delta t = T_e/200$.

Table 7.6 – Numerical configuration parameters for the KCS in head regular wave.

Numerical parameter	**Tested values**
`ddtScheme {default}`	`Euler`
`simulationType`	`RAS kOmegaSST`
`deltaT`	$T_e/200$

7.2.2 Meshes configuration

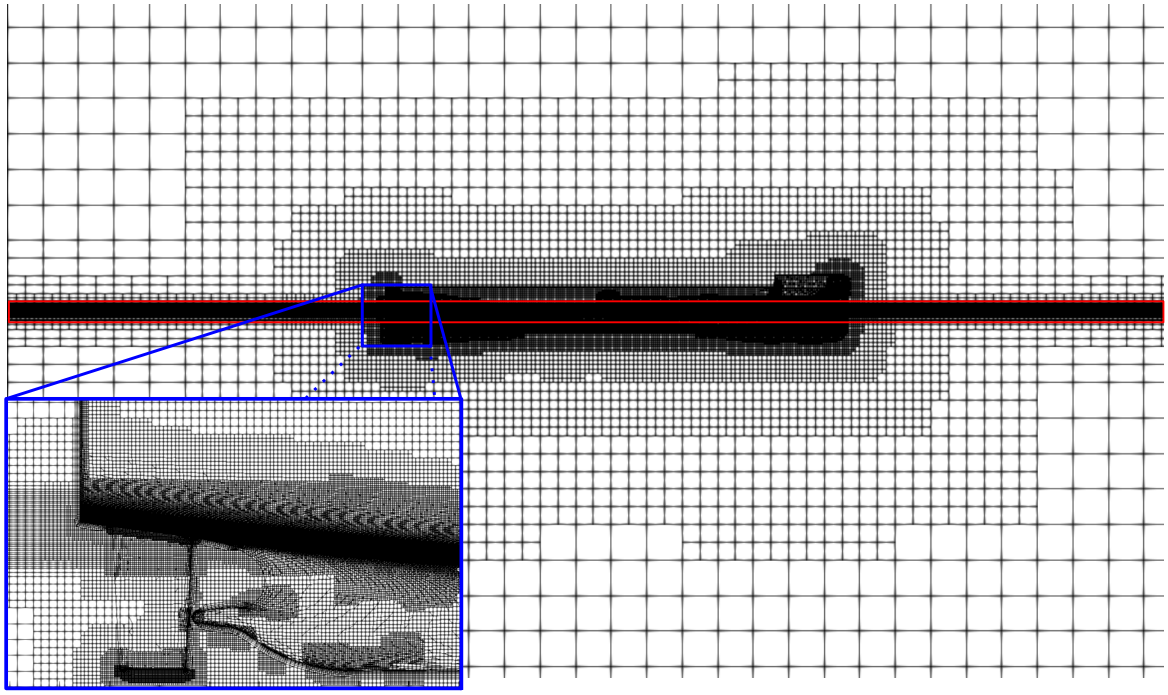


Figure 7.3 – Side view of the mesh at the symmetry plane and on the hull; KCS simulation; The red box identifies the free-surface zone. Blue box is a zoom on the stern of the ship.

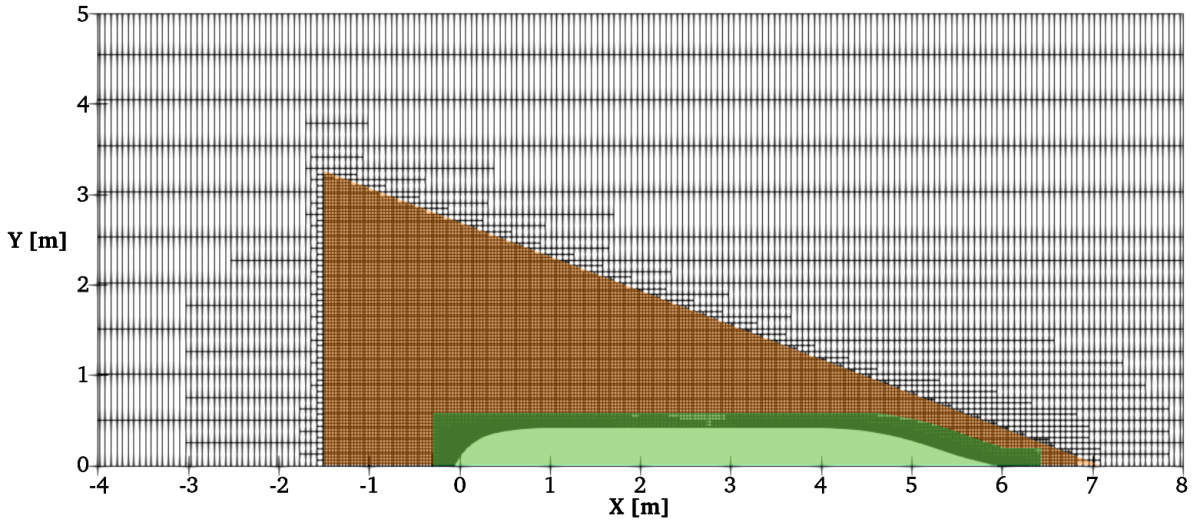


Figure 7.4 – Top view of the mesh at $z = 0$ m; KCS simulation; colors identify some refinement zones.

For this first assessment, the meshes are generated thanks to the Simcenter STAR-CCM+®. Figure 7.3 shows a side view of a the mesh configuration used. And Figure 7.4 shows a top view of the mesh at $z = 0$ m. First, a free-surface refinement zone is set with refinement depending on the wave condition (red box in Fig. 7.3). Then, a triangular refinement box is added in the free-surface zone close to the hull in order to simulate more accurately the wake (orange zone in Fig. 7.4). The near-field refinement is managed with some automatic refinement on the hull relatively to the surface curvature and adding some refinements near to the edges of the rudder, bow and transom. The view in the blue box in Fig. 7.3 shows the mesh aspect at the rear part of the ship. The diffusion of this hull refinement creates the refinement zone colored in green in Fig. 7.4. Finally, for the simulation of the turbulence, a wall function is applied, so a viscous layer with a first cell thickness $\Delta_{\mathrm{VL}} = 0.002$ m is inserted in the mesh.

A first study of the mesh convergence in calm water is done in order to identify an accurate mesh refinement near the body and over the domain. Then, based on a selected mesh density, the five meshes used for the five wave conditions are built.

Each resulting mesh is mainly composed of hexahedral cells, nevertheless, in order to improve the mesh alignment and the viscous layer insertion, other cell shapes can occasionally be generated.

a) Mesh convergence in calm water

Before building the meshes used for the simulation with waves, three mesh densities, "Coarse", "Medium" and "Fine" are tested in calm water in order to identify an accurate mesh refinement near the body and in the domain. Figure 7.5 shows a view of the mesh at the rear part of the ship for the "Coarse" (a), "Medium" (b) and "Fine" (c) meshes.

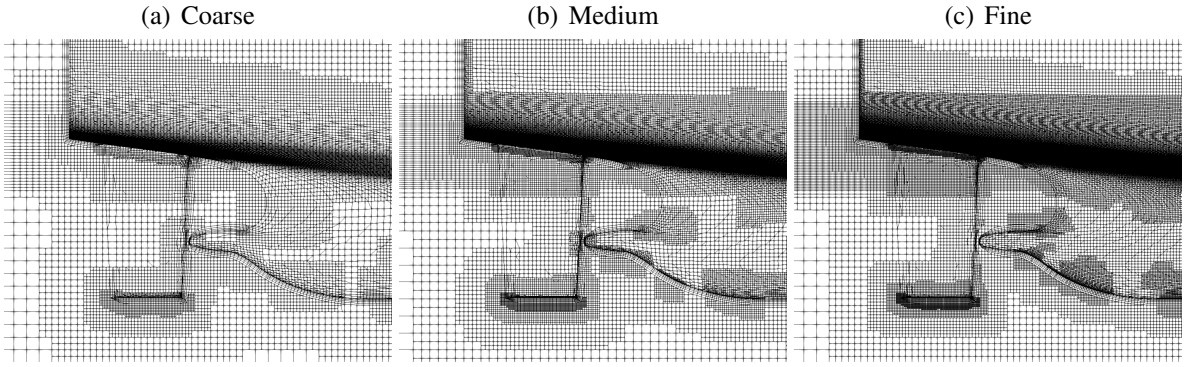


Figure 7.5 – Side views of the meshes used for simulations with calm water ; KCS; Fr=0.26

Table 7.7 gives some information on theses three meshes. The characteristic size of the cells are noted: Δx^{FF} for the far-field cells, Δx^{WF} for the wake-field cells and Δx^{NF} for the smallest near-field cells. The cell size of each zone is based on the far-field cell size Δx^{FF}, so for a given zone, the ratios $\Delta x^{\mathrm{FF}}/\Delta x$ are the same for the three meshes "Coarse", "Medium" and "Fine".

Table 7.7 – Mesh information for KCS in calm water cases

	$L_{PP}/\Delta x^{\mathrm{FF}}$	$\Delta x^{\mathrm{FF}}/\Delta x^{\mathrm{WF}}$	$\Delta x^{\mathrm{FF}}/\Delta x^{\mathrm{NF}}$	$L_{PP}/\Delta_{\mathrm{VL}}$	**Nb Cells**
Coarse	4	32	256	3035	1.7M
Medium	6	32	256	3035	3.7M
Fine	8	32	256	3035	6.7M

The converged results of the total resistance (R_T) coefficient $\left(\mathrm{C}_x = 2R_T/(\rho S_w U_0^2)\right)$ are shown in Table 7.8. The three meshes give results within 4% of the experiment. As a compromise between accuracy and mesh size, the medium mesh refinement is chosen as a base for the generation of every other mesh used to carry out the rest of the study.

Table 7.8 – Calm Water results comparison

	Coarse	**Medium**	**Fine**	**EFD**
$\mathbf{C_x \times 10^3}$	3.973	3.946	3.929	3.835
Err	+3.60%	+2.88%	+2.44%	

b) Meshes for simulations with regular waves

A set of meshes, is generated using the medium mesh density and adapting the free-surface refinement for each of the five regular wave conditions (Table 7.4). The free-surface box refinement is adapted to the wavelengths and wave heights which are considered. A synthesis of used meshes is provided in Table 7.9 in which Δx, Δy, Δz denote the dimensions of the cells in the free-surface zone.

The mesh refinement chosen at the free surface is motivated by the studies conducted by Li (2018) and Choi (2019). In order to propagate the waves using as well *foamStar* as *foamStar-SWENSE*, the criterion for the vertical refinement is around twenty cells per wave height and for the longitudinal refinement above fifty cells per wavelength.

Comparing to the recommendations made in Chapter 6, the ratio $\frac{\lambda \Delta z}{H \Delta x}$ is not between the prescribed bounds of 5 to 10. The free-surface cell aspect ratio ($\mathrm{AR_{FS}} = \Delta x / \Delta z$) is higher than the recommendations.

Table 7.9 – Mesh information for KCS in head regular waves

Mesh	$\lambda/\Delta x$	$\lambda/\Delta y$	$H/\Delta z$	$\frac{\lambda \Delta z}{H \Delta x}$	**Nb Cells**	**Related case**
M1	62	8	16	3.9	3.7M	**C1**
M2	82	10	20	4.1	3.7M	**C2**
M3	55	14	31	1.8	4.3M	**C3**
M4	66	16	19	3.5	3.4M	**C4**
M5	47	23	25	1.9	3.3M	**C5**

For this study, the meshes used for the simulations with waves are assumed to be consistent with the mesh used for the calm water simulation. So, in the following the total

resistance obtained from the calm water case is used to determine the added wave resistance for all the wave cases. This is a strong assumption and using the same mesh at least in a defined zone close to the ship might have be more relevant for the following analyses.

7.2.3 Regular head waves numerical results

a) Wave field damping

Because of the use of a first-order time scheme, a significant wave damping is noticeable. Figure 7.6 shows the wave elevation profile obtained for the case **C1** at the symmetry plane $y = 0$ (lower plot) and at a $y = L_{PP}$ (upper plot). This cutting planes are identified by dashed black lines in Figure 7.2. The profiles obtained using *foamStar* are colored in red and the one obtained using *foamStar-SWENSE* is colored in blue. The dark dashed line represents the wave profile obtained with the stream-function (`waveSF`) used to define the targeted field in the relaxation zones and at the boundaries of the domain. At $y = L_{PP}$, the wave are noticeably damped. This damping is less important with *foamStar-SWENSE* than with *foamStar*.

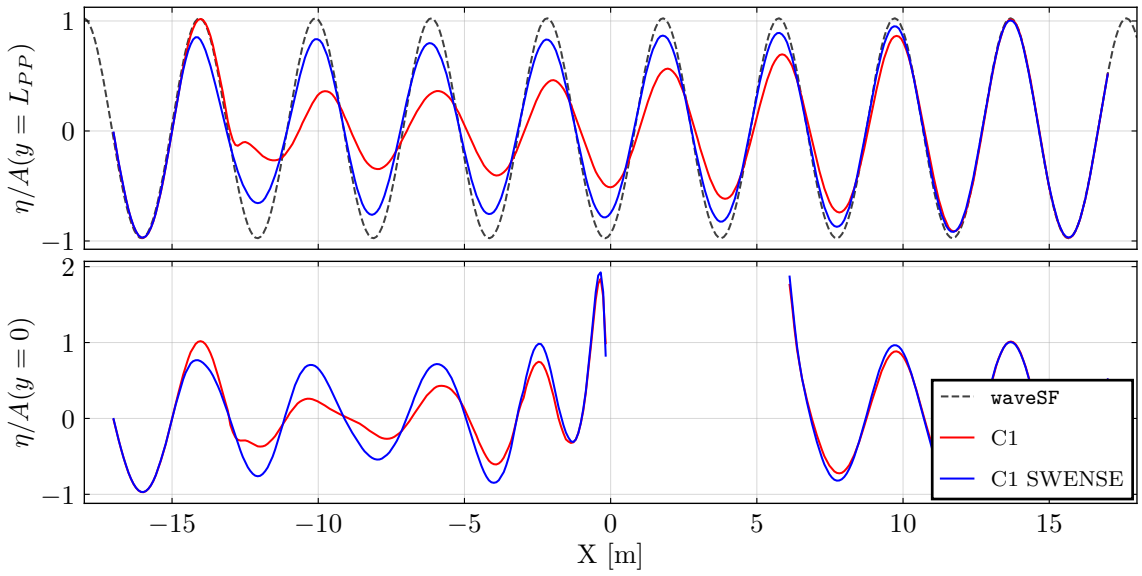


Figure 7.6 – Wave elevation profile at $y = L_{PP}$ and at the symmetry plane ($y = 0$); KCS Fr = 0.26; regular head wave $\lambda = 0.65 L_{PP}$; steepness $\varepsilon = 1.6\%$

Table 7.10 synthesizes the ratios H^{CFD}/H measured at the longitudinal location of the ship center of gravity (CoG) in the plane $y = L_{pp}$. H^{CFD} is the vertical distance between a crest and the trough of a wave that are equidistant from the CoG along the x-direction. Whatever

the wave condition the use of *foamStar-SWENSE* limits the wave damping. The shorter the wavelength, the more damped the wave is because the propagation distance is larger with respect to the wavelength.

Table 7.10 – Ratio H^{CFD}/H at the CoG

	C1	**C2**	**C3**	**C4**	**C5**
foamStar	0.60	0.66	0.75	0.80	0.84
foamStar-SWENSE	0.84	0.87	0.88	0.92	0.94

b) Forces and motions

The experimental data is provided in the form of RAOs computed with the 0^{th}, 1^{st} and 2^{nd} harmonics. To compute the same quantities with the numerical data, a spectral analysis is conducted on the total resistance, heave and pitch time histories for the five wave conditions studied. In the present section the wave amplitude A used for the nondimensionalization is the prescribed wave height H divided by two for the EFD and half of the measured wave height at the CoG for the CFD results. Because of the significant damping noticeable in *foamStar* simulations, this choice could be discussed.

The 0^{th}, 1^{st} and 2^{nd} harmonic motion RAOs are respectively presented in Fig. 7.7. The mean and first harmonic amplitude of the heave and pitch motions obtained with *foamStar* and *foamStar-SWENSE* reveal some differences between the two solvers. The motion amplitude seems to be slightly smaller for the simulations done with *foamStar*, especially close to the resonance (case **C3**). This difference might be induced by the significant damping previously mentioned. Nevertheless, with the used nondimensionalization, both *foamStar* and *foamStar-SWENSE* provide resuslts close to the experimental data. On Fig. 7.7 (e) and (f) it can be seen that also the motion second harmonic amplitudes computed with *foamStar* and *foamStar-SWENSE* are close to each other and follow the experimental data. The major difference between both remains the undervalued motion amplitude with *foamStar* for the case **C3**.

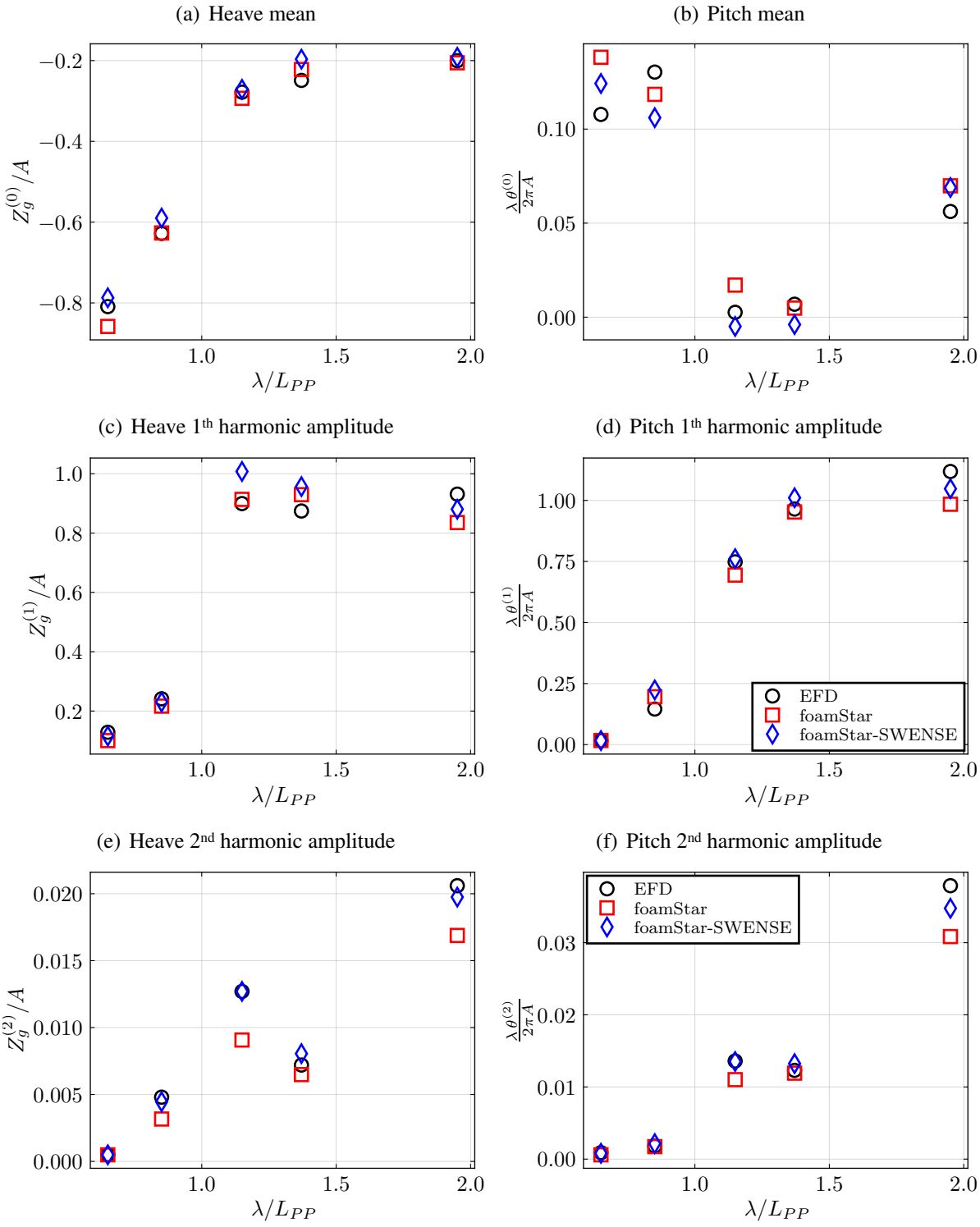


Figure 7.7 – KCS heave and pitch 0th, 1st and 2nd harmonic amplitude, Fr=0.26, head waves

In Fig. 7.8, the resistance is also compared with experimental results in terms of total resistance (a) and added resistance (b). The total resistance coefficients computed with *foamStar* and *foamStar-SWENSE* are close to the experimental results with still the undervalued coeficient with *foamStar* simulating **C3**. The added resistance values computed with *foamStar* and *foamStar-SWENSE* are quite different for short wavelengths. For those conditions the added resistance magnitude is very small with respect to the calm water resistance and the uncertainty on the added resistance computation is high. The noticed damping is likely also involved. The numerical values of the total resistance and differences relative to the experimental data are detailed in Table 7.11. The first and second harmonics of the total resistance time signal are presented in Fig. 7.8 (c) and (d), respectively. The first and second harmonic analysis confirms the closeness of the *foamStar* and *foamStar-SWENSE* results and the general agreement with the experiments.

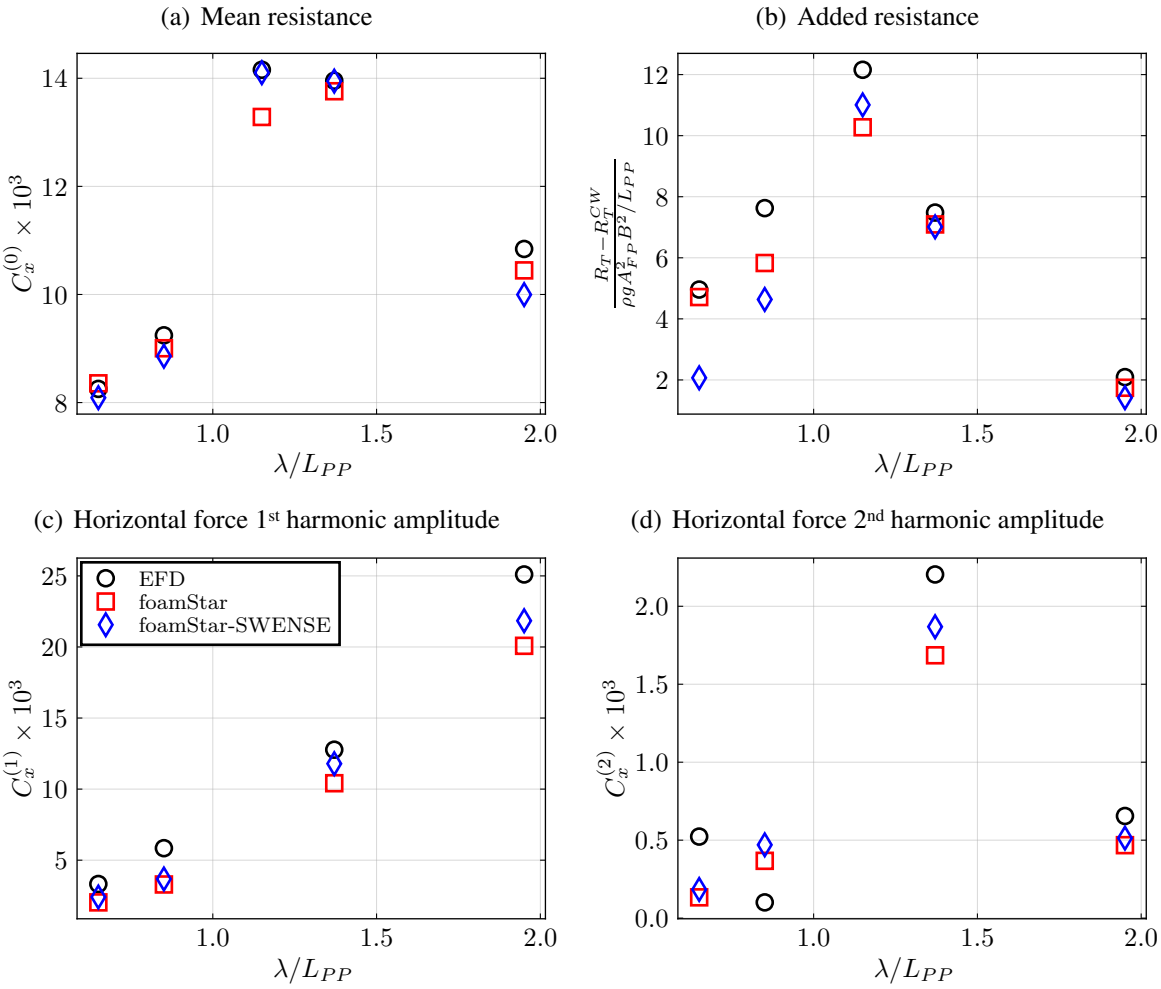


Figure 7.8 – KCS total resistance and added resistance, Fr=0.26, head waves

Table 7.11 – KCS Total resistance results, (FSS) exponent stands for *foamStar-SWENSE* and (FS) exponent for *foamStar* solver

Case	$C_x^{(EFD)} \times 10^{-3}$	$C_x^{(FSS)} \times 10^{-3}$	$\frac{C_x^{(FSS)} - C_x^{(EFD)}}{C_x^{(EFD)}}$	$C_x^{(FS)} (10^{-3})$	$\frac{C_x^{(FS)} - C_x^{(EFD)}}{C_x^{(EFD)}}$
C1	8.25	8.09	-2.0%	8.36	1.2%
C2	9.24	8.86	-4.2%	9.00	-2.5%
C3	14.16	14.10	-0.4%	13.28	-6.2%
C4	13.96	13.95	-0.1%	13.76	-1.4%
C5	10.84	10.00	-7.8%	10.45	-3.7%

The wave elevation field obtained from *foamStar* and *foamStar-SWENSE* for the C4 condition is presented in Fig. 7.9. The wave patterns are globally similar but some differences can be observed, *foamStar-SWENSE* magnitudes being a bit larger.

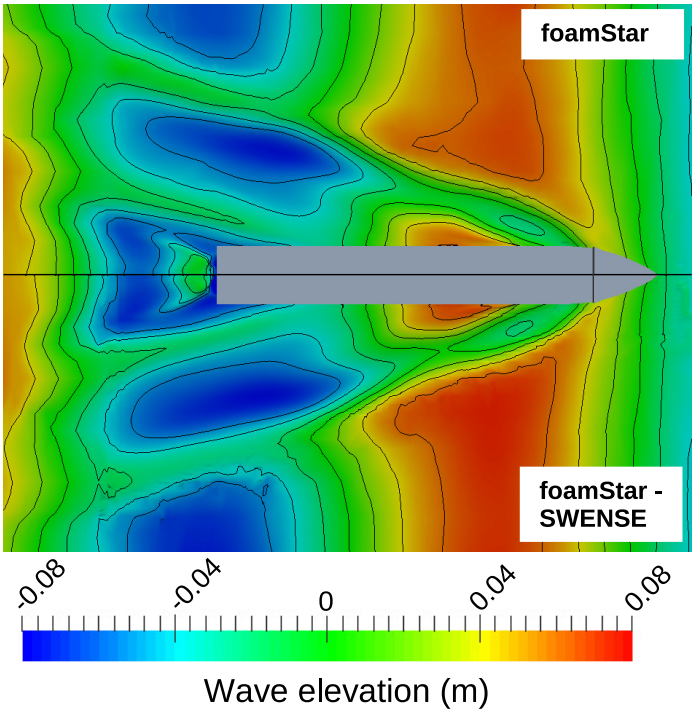


Figure 7.9 – Comparison of the KCS wake in head wave (C4) using *foamStar* and *foamStar-SWENSE*

7.3 Ship in long crested irregular waves

In the work presented during the 33rd Symposium on Naval Hydrodynamics, a study of the capability of *foamStar-SWENSE* was also carried out on one rough irregular sea sate with forward speed. The chosen sea state is classified with the code six in the World Meteorological Organization. The chosen sea state follows a Pierson-Moskowitz spectrum, the characteristics of which are shown in the Table 7.12.

Table 7.12 – Irregular sea state (SS6) characteristics

Scale	$\mathbf{H_s}$ [m]	$\mathbf{T_p}$ [s]	$\mathbf{H_s/\lambda_p}$
Full	5.0	12.4	0.021
Model	0.132	2.0145	

The KCS model scale and the mesh generation strategy are the same as for the previous study in regular waves. The free surface refinement is adapted with 33 cells per significant wave height (H_s) and 100 cells per peak wavelength (λ_p). The time step is also adapted so that 500 time steps are computed per encounter peak period ($T_e = 1.23$ s).

This simulation is conducted using the same degrees of freedom and forward speed conditions as for the study in regular waves (Table 7.2 and Table 7.3). Figure 7.10 shows the interaction between the wake generated by the ship and the incident waves. Figure 7.11 shows time histories for the heave and pitch responses as well as the total resistance. These preliminary results show the stability and the ability of the recent *foamStar-SWENSE* implementation to compute such seakeeping problems. The duration of presented simulation was to short to conduct some spectral and stockastic analysis for instance measuring the heights probability of exceedance. Because of the weak accuracy of the first order Euler time scheme, before carrying out more studies with ship in irregular sea states, some improvements such as the one presented in the following are needed.

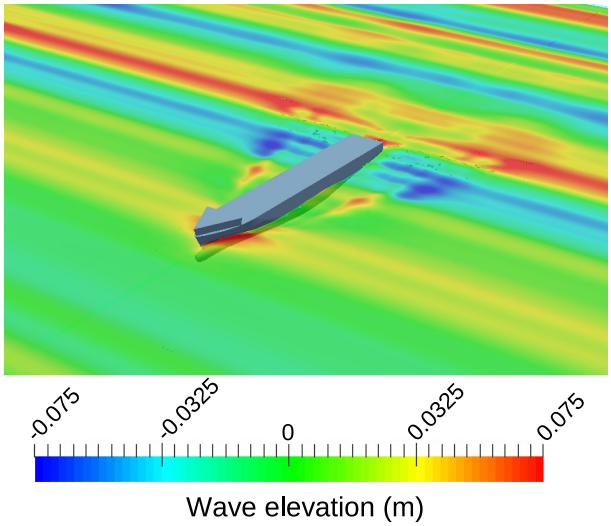


Figure 7.10 – KCS wake in irregular head wave (SS6), Fr=0.26

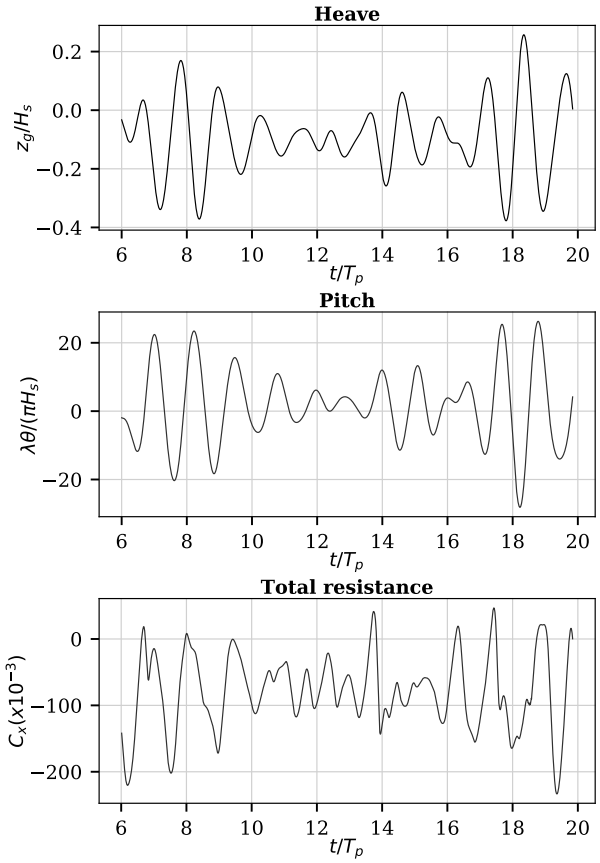


Figure 7.11 – KCS motions (heave and pitch) and total resistance response in irregular head wave (SS6), Fr=0.26

7.4 Application of the recommendations

The previous study was done before establishing some recommendations for the mesh generation and numerical setup. The objective is now to apply these recommendations in order to improve the accuracy of the simulations. To this aim, case **C1** is considered in this section, as its wave characteristics and the size of the simulation domain are close to the studied ones in Chapt. 4. Both the mesh and the numerical setup are changed compared to the simulation made previously in Sec. 7.2.

7.4.1 Mesh configuration

First, in Chapt. 4 it is shown that the mesh characteristics influences a lot the accuracy of *foamStar*. The skewness of the cells seems to be the most problematic one. It was also shown

that using a cell skewness corrector improves the solver accuracy in case of some skewed cells. However, some trials were made and no simulation with ship presence was successfully run with this skewness corrector. The reason is not yet identified, so it was decided not to use this corrector but to limit the skewness thanks to the use of *snappyHexMesh* program. This mesh generator is dedicated to OpenFOAM simulation, so it allows a better control of some mesh characteristics such as the skewness.

Then, the mesh characteristics are built following the recommendations made in Chapt. 6. A background mesh is firstly built, with these recommendations in mind and then *snappyHexMesh* is used to add cell refinements close to the hull and to snap the cells to it. Figure 7.12 shows a view of the background mesh with the prescribed refinement zones and a zoom on the refinements applied close to the ship. The sizing of the cells in each zone (from 0 to 6) follows the method described in Sec. 5.3. As for the meshes generated using Simcenter STAR-CCM+, additional refinements are made close to the edges of the rudder, bow and transom. Figure 7.13 shows a top view of the mesh at $z = 0$ m. The wake refinement is made using two boxes near to the hull (orange and yellow colored). The refinement zone close to the hull is colored in green on the figure.

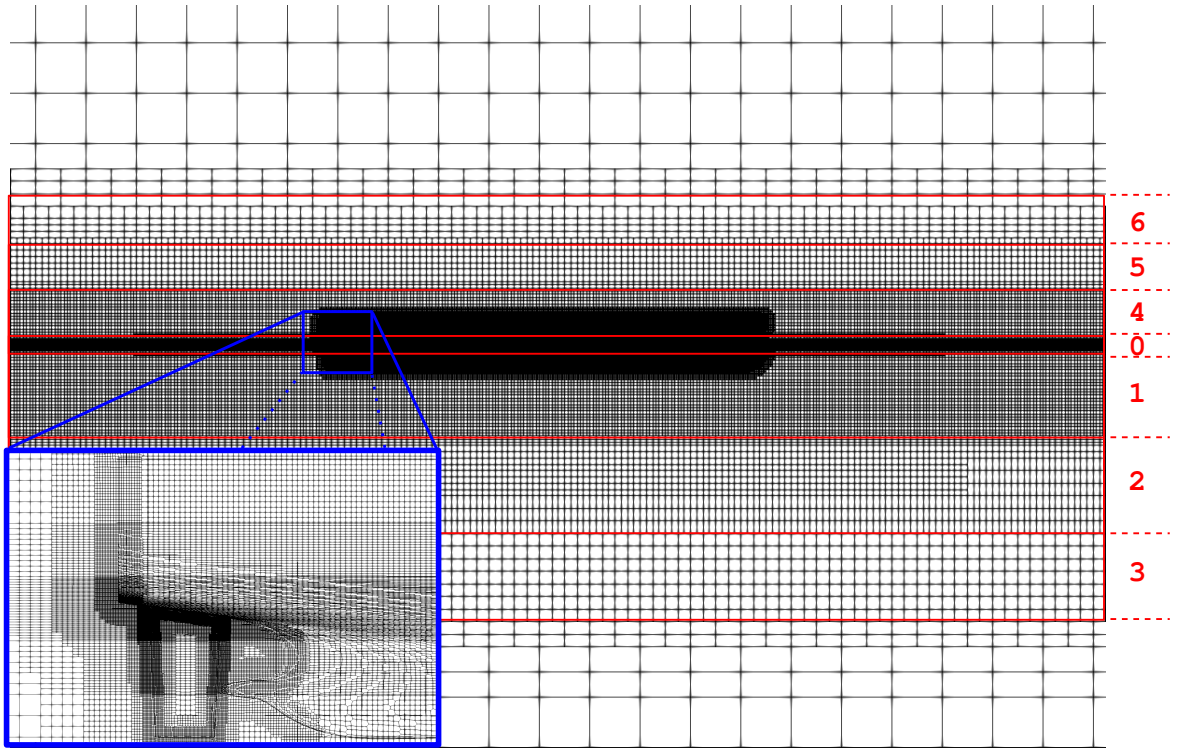


Figure 7.12 – Side view of the improved mesh at the symmetry plane and on the hull; KCS simulation; The red box identifies the free-surface zones. Blue box is a zoom on the stern of the ship.

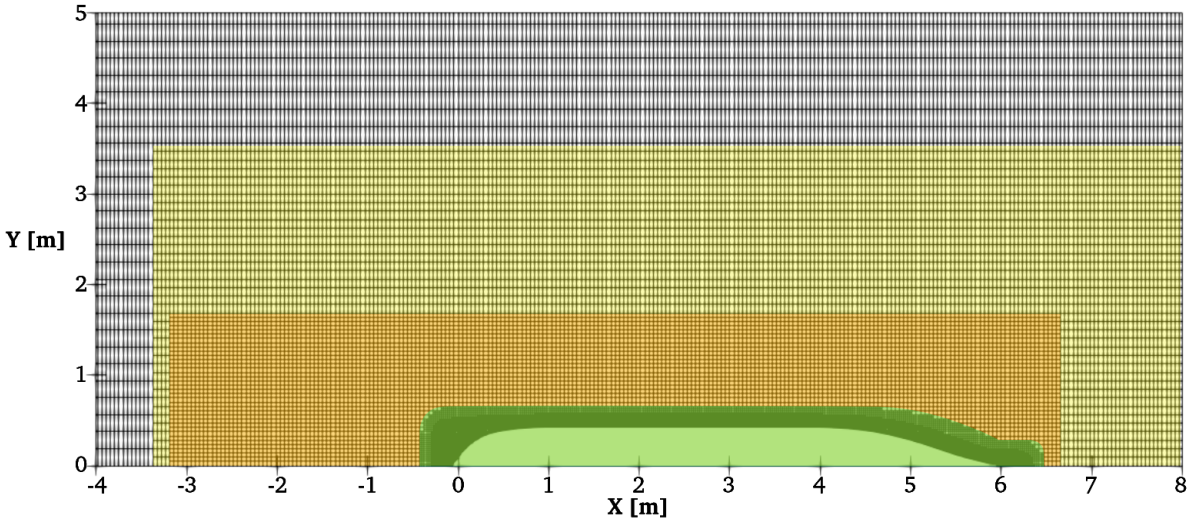


Figure 7.13 – Top view of the improved mesh at $z = 0$ m; KCS simulation; colors identify some refinement zones.

7.4.2 Settings

In this section the numerical set-up is the reference one detailed in Sec. 3.6. Three different time schemes are tested: implicit Euler, second-order backward and Crank-Nicolson 0.95, and the turbulent model used is fs-k-ω-SST. Table 7.13 gives a synthesis of these parameters.

Following the recommendations, no inlet relaxation zone is used. Figure 7.14 shows the used configuration.

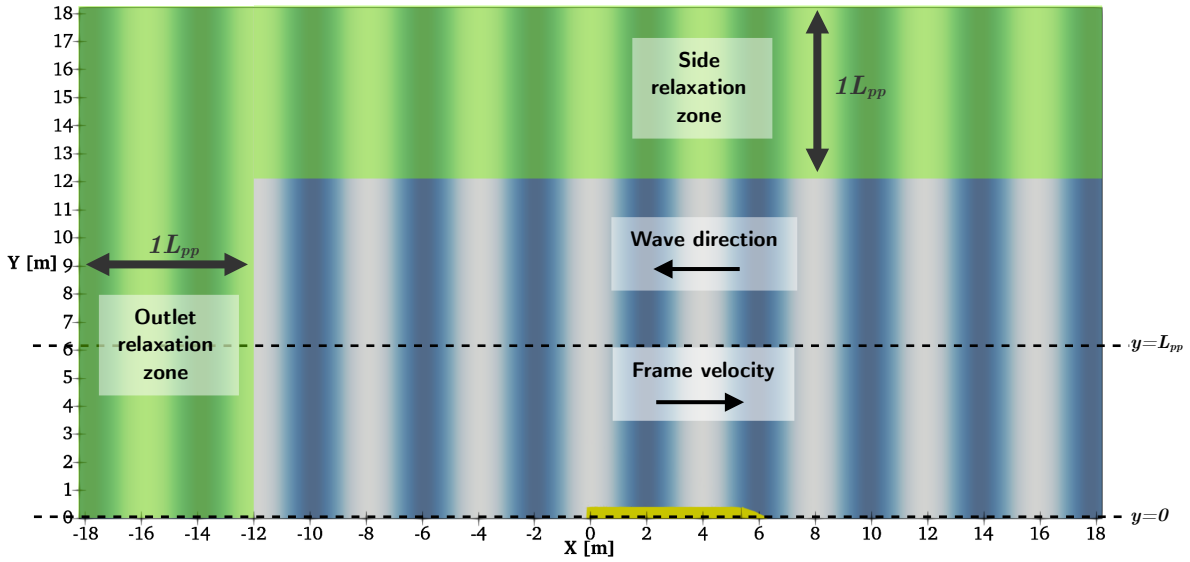


Figure 7.14 – Relaxation zone configuration for KCS simulations without inlet forcing zone

Table 7.13 – Improved numerical configuration parameters for the KCS in head regular wave

Numerical parameter		**Tested values**

`ddtScheme`	`ddt(U)`
	`ddt(rho,U)`
		`Euler`
		`CranckNicolson 0.95`
		`backward`

	`ddt(rhoPhi,omega)` `ddt(rhoPhi,k)`	`Euler`
`simulationType`		`RAS kOmegaSST`
`deltaT`		$T_e/200$

7.4.3 Results

No added resistance or RAO analyses are done in this section. The objective here is to verify the ability of the new setup to run second-order time schemes and then to observe the differences in the wave field and time traces when using Euler (`EULER`), second-order backward (`BCK`) and Crank-Nicolson 0.95 (`CN95`).

a) Wave field damping

Figure 7.15 shows the wave elevation profile obtained at the symmetry plane $y = 0$ (lower plot) and at $y = L_{PP}$ (upper plot). The profiles obtained using `EULER`, `BCK` and `CN95` are colored in blue, red and green, respectively and the dark dashed line represents the wave profile obtained with the stream-function (`waveSF`) used to define the targeted field in the relaxation zones and at the boundaries of the domain. At $y = L_{PP}$, the waves are significantly damped with `EULER` compared to the waves obtained with `BCK` and `CN95`. Still at $y = L_{PP}$, the profiles with `BCK` and `CN95` are close to each other, some wave amplitudes are larger than the analytic solution and a non regular profile is noticeable at the starting of the damping zone. 3D simulations with a similar mesh but without ship should be done to investigate which of these perturbations are due to the ship and which are not. For the wave-elevation profiles at the symmetry plane, the `EULER` scheme shows damping and the profiles with `BCK` and `CN95` are still close to each other.

Table 7.14 synthesizes the ratios H^{CFD}/H measured at the longitudinal location of the ship center of gravity (CoG) in the plane $y = L_{pp}$, with this improved configuration. With both `CN95` and `BCK` this ratio remains close to one so the damping can be assumed to be negligible. A major improvement is thus obtained compared to the simulation results of the first assessment made at the beginning of the thesis (see Tab. 7.10 in Sec. 7.2). Using `EULER`, the damping of the wave elevation is about 32% that is comparable with the 30% damping observed with the configuration used in that first assessment (Tab. 7.10).

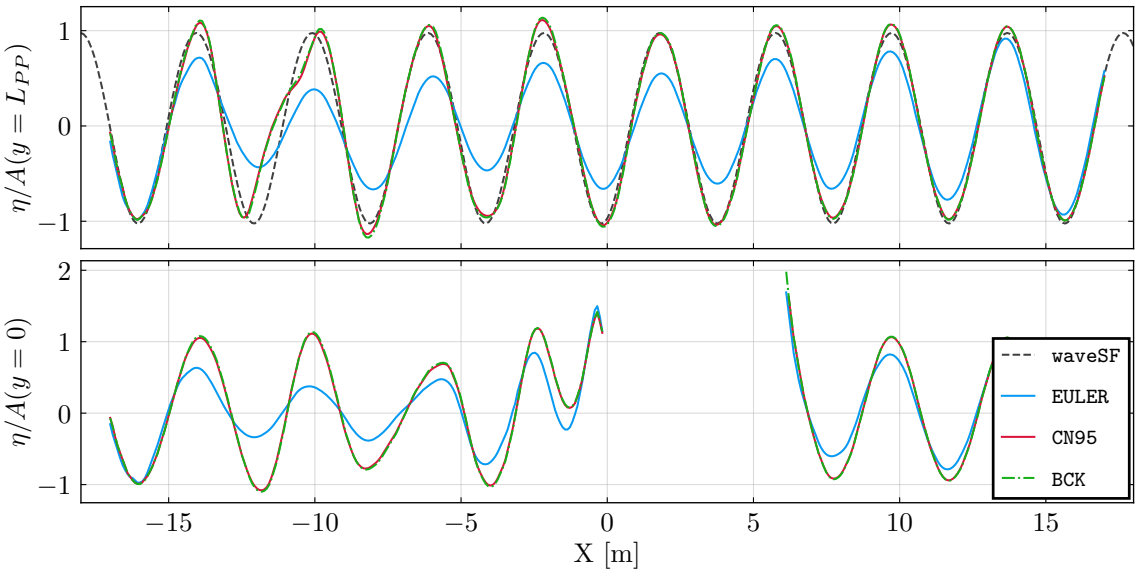


Figure 7.15 – Wave elevation profile at $y = L_{PP}$ and at the symmetry plane ($y = 0$) using Euler, second-order backward and Crank-Nicolson 0.95 time schemes; KCS Fr = 0.26; regular head wave $\lambda = 0.65 L_{PP}$; steepness $\varepsilon = 1.6\%$; Improved configuration.

Table 7.14 – Ratio H^{CFD}/H at the CoG in the case **C1** using the improved configuration

	EULER	**CN95**	**BCK**
$\boldsymbol{H^{CFD}/H}$	0.577	1.004	0.997

Figure 7.16 shows the top views of the wave elevation map obtained simulating with Euler and second-order backward time schemes at $t = 20T_e$. Dark lines represent the iso-wave-elevation with $\eta \in \{-0.1, -0.09, -0.08 \; .. \; 0.09, 0.1\}$ The damping of the wave field with the Euler scheme is clearly visible. Figure 7.17 shows the same view but using Crank-Nicolson 0.95 rather than Euler. The wave elevation obtained with `BCK` and `CN95` are really close to each other but some slightly steeper wave field can be identified with `BCK`.

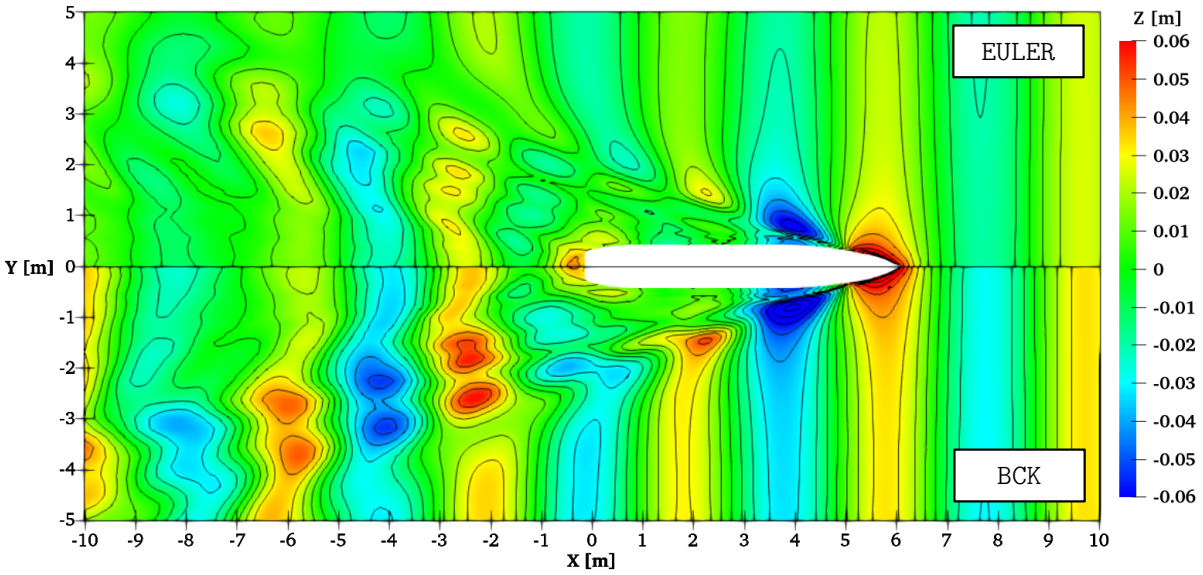


Figure 7.16 – Comparison of the wave elevation map obtained using Euler and second-order backward time schemes at $t = 20T_e$; KCS Fr = 0.26; regular head wave $\lambda = 0.65L_{PP}$; steepness $\varepsilon = 1.6\%$

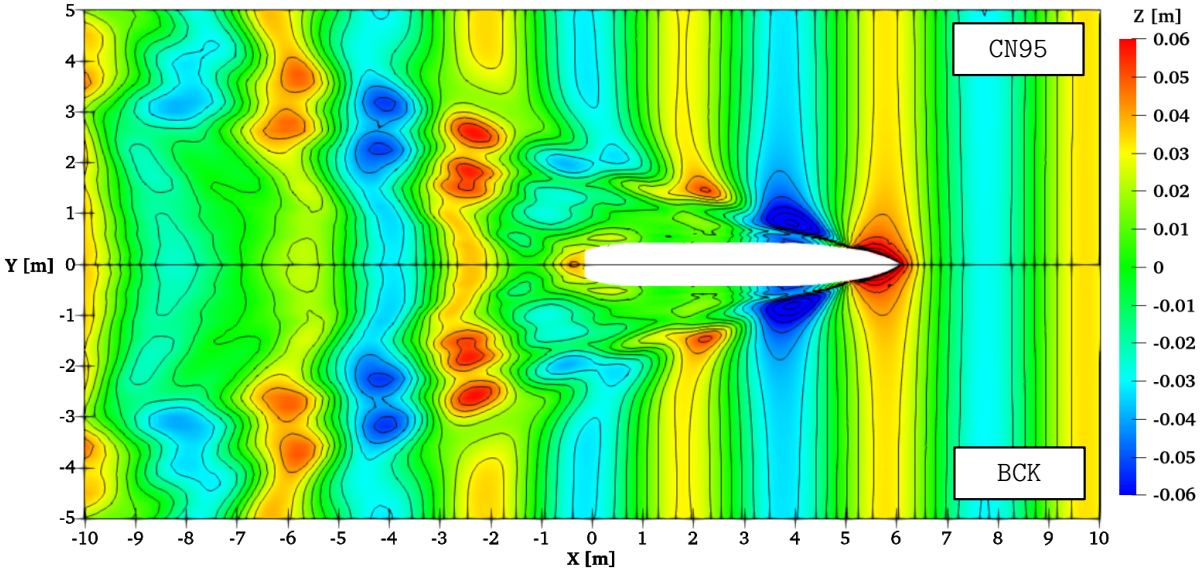


Figure 7.17 – Comparison of the wave elevation map obtained using Crank-Nicolson 0.95 and second-order backward time schemes at $t = 20T_e$; KCS Fr = 0.26; regular head wave $\lambda = 0.65L_{PP}$; steepness $\varepsilon = 1.6\%$.

b) Forces and motions

The time trace of the non-dimensional longitudinal pressure and viscous forces are plotted on Fig. 7.18 and the time trace of the non-dimensional heave and pitch motions in Fig. 7.19. `EULER`, `CN95` and `BCK` are represented in blue, red and green, respectively. At a given time t, the mean forces are calculated using a three encountered period long sliding window between $t_0 = t - 1.5$ s and $t_1 = t + 1.5$ s. These mean forces are represented with dotted lines of the same color as the corresponding simulations.

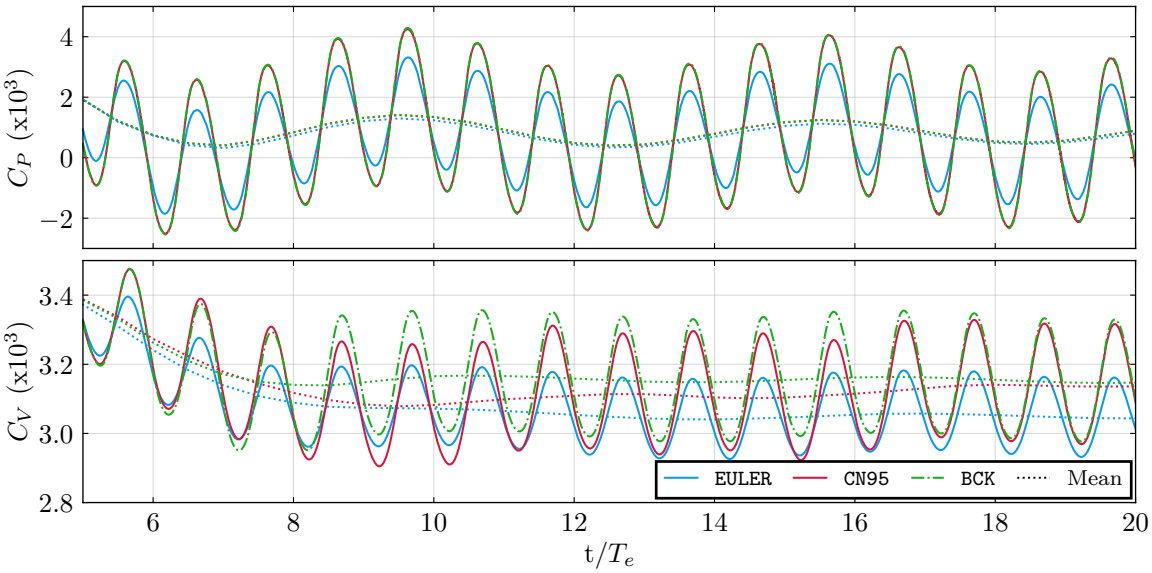


Figure 7.18 – Time history of the pressure resistance and viscous resistance coefficients using Euler, second-order backward and Crank-Nicolson 0.95; KCS Fr = 0.26; regular head wave $\lambda = 0.65 L_{PP}$; steepness $\varepsilon = 1.6\%$.

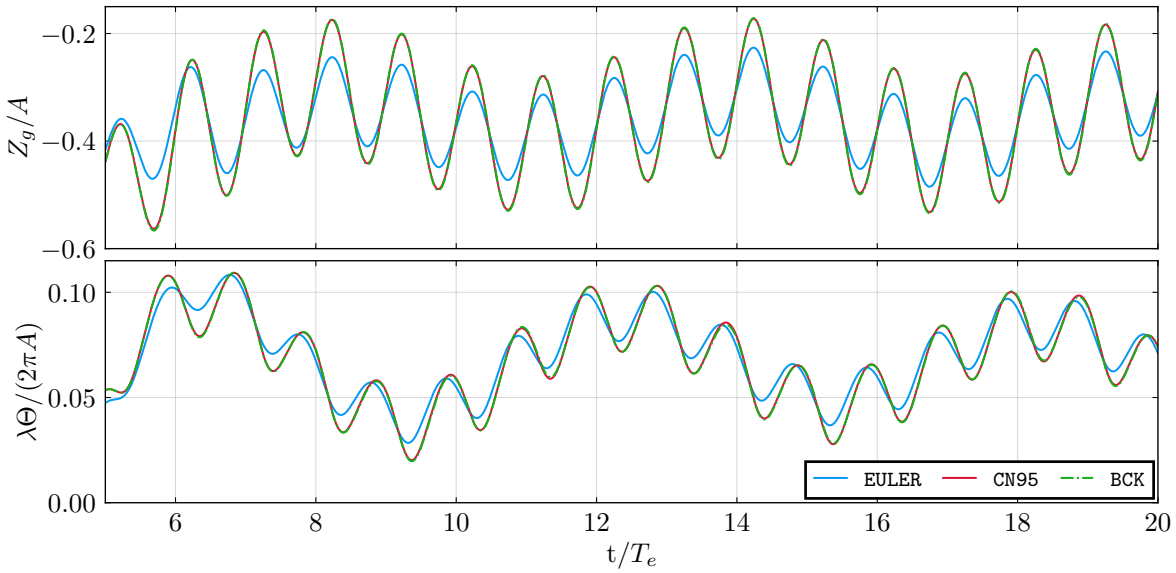


Figure 7.19 – Time history of the heave and pitch using Euler, second-order backward and Crank-Nicolson 0.95; KCS Fr = 0.26; regular head wave $\lambda = 0.65 L_{PP}$; steepness $\varepsilon = 1.6\%$.

First, looking at the longitudinal forces (Fig. 7.18), the amplitude of the forces with `EULER` are smaller than the ones obtained with `CN95` and `BCK`. This is due to the smaller wave amplitude of the waves around the ship with `EULER` time scheme that is itself due to the the numerical damping. The time traces obtained with `CN95` and `BCK` are superimposed for the pressure force but differ for the viscous force. The viscous force obtained with `CN95` is below the `BCK` one during several periods before converging progressively to the same value. The results from the `BCK` simulation can be assumed as converged after 10 periods when the simulation with `CN95` takes about 18 period to converge. With `EULER`, the viscous force also converges within 10 periods but to a value 3% smaller than the converged value of `BCK` and `CN95` simulations. To explain these differences, further investigation are needed. This viscous force depends strongly on the mesh quality close to the hull and on the size of the first mesh layers. This aspect is not investigated in this section. It can be noticed that the mean pressure forces are really close to each other regardless of the scheme used.

Besides, looking at the motions, the amplitudes of both heave and pitch motions obtained with `EULER` are smaller than the ones obtained with `CN95` and `BCK`. Moreover, with `EULER` on one side and `BCK` and `CN95` on the other side, a phase shift of the pitch motion is noticeable. The wave damping explains probably most of these differences and investigations of the evolution of the motions depending on the prescribed wave height should be carried out.

No differences can be noticed from the motion time traces between `CN95` and `BCK`.

7.4.4 Overview of the simulation improvements

First, applying the new recommendations, simulations using the Crank-Nicolson 0.95 time scheme have been run successfully, whereas using the configuration tried at the beginning of this thesis, simulations with Crank-Nicolson 0.95 were not stable and used to fail. The free-surface cells aspect ratio, the location and the size of the refinement boxes, or the use of Simcenter STAR-CCM+ non-taking into account some OpenFOAM mesh requirements could be reasons explaining the troubles encountered with the firstly tried configuration and Crank-Nicolson 0.95. Some improvements on the parameters choice are also likely to be involved.

Besides, the new second-order backward MULES version seems to show consistent results compared with the Crank-Nicolson MULES so it corroborates the validity [1] of its implementation.

7.5 Conclusion

The first assessment presented in this section reported on the first results obtained at the beginning of the thesis, and based on some developments in the *foamStar-SWENSE* solver presented by Li (2018). This study has been the first application of *foamStar-SWENSE* toward the resolution of naval seakeeping problems with forward speed and has been completed by Kim (2021). As a first approach these Z. Li's developments appeared to be able to reach the objective.

In this study, the comparison of results obtained with *foamStar* and *foamStar-SWENSE* on the same meshes does not highlight the capabilities of the SWENSE method but allows to assess its implementation.

It must be verified that the *foamStar-SWENSE* solver is able to keep a good accuracy for lighter meshes in the far-field region. This will be the next step together with further numerical improvements in the implementation of high-order discretisation schemes.

As shown, some numerical parameters, domain configuration and meshes used in this first assessment did not correspond to the recommendations made in Chapter 6. A first application of the newly implemented schemes and setup recommendations has been done at

1. Here "validity" does not refer to "Validation" from V&V procedures

the end of the thesis and is reported in the last part of this chapter. It shows improvements compared to the first assessment. This shall be extended with future works through a more detailed mesh convergence analysis and applying V&V procedures.

CHAPTER 8

APPLICATION IN THE INDUSTRIAL CONTEXT

One of the main objectives of this Ph.D. is to meet the expectations of BVS-M&O regarding the optimization of naval simulations in waves. To reach this objective, the study of the accuracy of the *foamStar* solver, the implementation of optimized methods and building up a methodology are required. In order to keep close to this industrial objective, during this Ph.D., I brought my help on the first industrial applications of *foamStar* at BVS-M&O. Some applications were done this year by the company and Hein Kyaw Swa, an intern at BVS-M&O, has started to work on the validation of *foamStar* on several industrial cases close to the one studied in this Ph.D. This section describes the work done about the integration of *foamStar* into the industrial context.

8.1 Quick integration with applicable guidelines

The first issue with the use of *foamStar* in an industrial context with several co-workers is to form them to the use of the solver. From a research perspective, an OpenFOAM based code has the strong advantage to give access to a very large amount of numerical schemes and parameters and a lot of flexibility. Nevertheless, this becomes detrimental when it comes for a company to use the solver on tested and validated cases with well defined recommendations. In this situation the flexibility of the solver slows the handling and turns into a source of errors.

Consequently the strategy chosen in order to ease the industrialization was as follow:

1. To identify the fields of application of *foamStar* and define recommendations for some specific cases.
2. To provide a simplified set of modifiable parameters.
3. To provide some user guides.

8.1.1 Fields of application and recommendations

This first step is based on the sharing of the conclusions established by the researchers working and having worked on the software with the members of BVS M&O.

Wave generation This has been studied by several Ph.D. students at the LHEEA in recent years. Li (2018), Choi (2019), Kim (2021) and the work developed in Chapter 6 provide some recommendations on the current optimal setup for pure wave propagation with regular and irregular waves.

Naval simulation in calm water A part of the work in Chapt. 7 has led to the conclusion that *foamStar* can be applicable to resistance calculation in calm water with forward speed.

Naval simulation in waves The work done on the Tokyo 2015 case 2.10 (Sec. 7.2) has led to the conclusion that both the *foamStar-SWENSE* and the standard *foamStar* (RANSE) can eventually be used for naval simulations. Kim (2021) also provides similar conclusions. For short term simulations with fixed bodies the SWENSE method also appears to be more efficient than the RANSE one (Li (2018), Li et al. (2021) and S. Aliyar current Ph.D.). However, for complex geometry, forward speed and non homogeneous refinements, the current implementation of SWENSE turns out to be unstable. The preliminary simulations reported in Chapter 7 and in Kim (2021) show the feasibility of naval simulation with forward speed in head long-crested irregular waves using *foamStar-SWENSE* but further and more systematic investigations are still to be done before the introduction of it in the industrial process.

8.1.2 Simplified setup

The creation of a simplified setup with a reduced number of parameters encompasses several objectives. The first one is to speed up the handling of *foamStar* by the BVS M&O members. The second one is to prevent the modification of parameters that have been considered as the optimal ones.

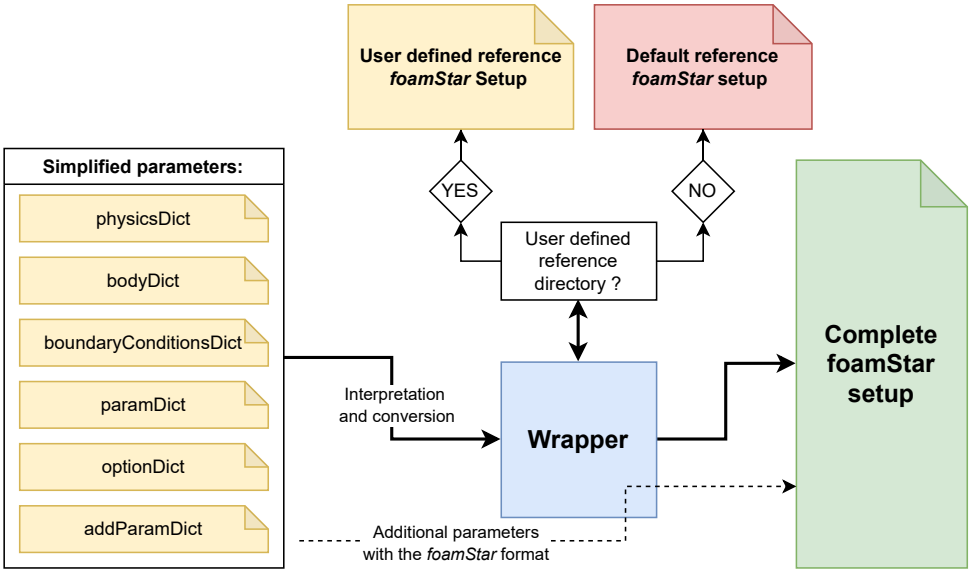


Figure 8.1 – BVS M&O *foamStar* setup procedure

Figure 8.1 illustrates the procedure developed for the use of *foamStar* at BVS M&O. At the center of the procedure, a script is used to read the simplified set of parameters. This set is made to be close to the set of parameters usually modified by BVS M&O in other industrial solvers such as ISIS-CFD and STAR-CCM+. It was important to be as close as possible to the parameters the members of the company are used to manipulate in terms of names, significations, and organization but without hiding the *foamStar* particularities. Then the script identifies the simulation type required by the user (Symetric simulations, multiphase, calm water etc ...) and uses reference *foamStar* parameters that have been fixed according to the established recommendations. The output is the complete *foamStar* setup and running scripts can be launched.

Simplified parameters description:

physicsDict *In this file are indicated the physical conditions in the domain and the wave fields.*

bodyDict *In this file are indicated the bodies and the imposed motion of the whole domain.*

boundaryConditionsDict *In this file are indicated the boundary-conditions types.*

paramDict *In this file are indicated some parameters of the computation.*

optionDict *In this file are indicated the additional options (for instance Actuator disk).*

addParamDict *In this file are written all the additional parameters that we want to add or modify in relation to the reference case.*

8.1.3 User guides

Several user guides or presentations have been made in order to circulate them inside BVS M&O. Special attention has been paid to the mesh generation process and mesh generation recommendations have been established. The mesh generation is a crucial step of CFD but it is above all the case with *foamStar* that is strongly sensitive to mesh quality. Those recommendations have been tested by Kyaw Swa Hein.

8.2 Computational cost managment through the control of the residuals

Aside from the numerical optimization of the code, the choice of convenient numerical parameters is crucial for the computational cost. One of the key parameters is the number of PIMPLE and PISO iterations. Imposing more PIMPLE iteration implies more VOF, motion (in case of moving body) and PISO resolutions. As seen in Section 3.3, each operation is related to one or several time-consuming linear systems resolutions. Increasing the number of PISO iterations increases the number of times the pressure equation is solved (Eq. 3.119), this being the more costly operation in the code.

The duration of one linear system resolution depends on the difference between the known approximation of the solution and the solution itself. This difference is measured with the initial residual (more details can be found in Sec. 3.3.7 and Sec. 3.4). If this initial residual is close to the chosen absolute tolerance (ρ_{err}) of the system resolution, only few iterations are needed. However, when this initial residual is high the linear solvers require a large number of iterations before reaching the expected tolerance.

At the m$^{\text{th}}$ PIMPLE iteration in the n$^{\text{th}}$ time step, the convergence is optimal if a m+1 PIMPLE iteration would not change significantly the matrix systems. In other terms, whatever the solved variable ψ and its associated system $\mathbf{A}\boldsymbol{\Psi} = \mathbf{B}$, the residual of the system verifies:

$$r_{\psi}^{(\text{n,m+1,0})} := Res\left(\mathbf{A}^{(\text{n,m+1})}, \mathbf{S}^{(\text{n,m+1})}, \psi^{(\text{n,m,q}_{\text{last}})}\right) < \rho_{\text{err}}$$

Getting an initial residual below the tolerance, the linear system is already assumed as "solved", so no solver iteration is needed. The computational cost of such an m+1 PIMPLE iteration is reduced to the matrices building and additional numerical corrections, so adding unnecessary PIMPLE iteration does not increase a lot the computational cost.

Because of the spatial discretization process (schemes choice, mesh structure etc ...), for some fields the initial residual at the beginning of a PIMPLE iteration may saturate and it implies costly linear resolutions at each PIMPLE loop without reducing the apparent iterative error. An example of such a situation is the simulation presented in Section 7.4. For this simulation, in order to ensure the convergence and to minimize the iterative error, 5 PISO iterations within 10 PIMPLE iterations are used and the absolute tolerance is set to 10^{-10}. Figure 8.2 shows the initial pressure residual coming from the resolution of Eq. 3.119. The GAMG solver is used for the linear system resolution and the horizontal axis represents the the total number of GAMG iterations from the beginning of a given time step. The beginning of a given time step is identified by a red stem with a height equal to the initial residual at this step of the simulation. Yellow stems identify the beginning of each PIMPLE iterations. Blue bars represent the initial residual of PISO iterations (height of the bar) and the number of linear solver iterations used all along a given PISO iteration (width of the bar). The green bars represent the initial residual and the total number of linear solver iterations for a given PIMPLE loop.

In Fig. 8.2, the initial residual saturates above 10^{-2}, so each additional PIMPLE iteration in the saturation zone ($4 \leq$ PIMPLE iter. ≤ 10) requires about the same number of GAMG iterations. The width of the bars are proportional to the number of GAMG iterations, so a same number of GAMG iterationsfor two PIMPLE iterations means a similar computation cost. Consequently, assuming that the pressure residual is a relevant indicator of the iterative error, the figure shows that the computational cost of the time step might have been reduced by a factor 2 or 3 removing some apparently unnecessary PIMPLE iterations.

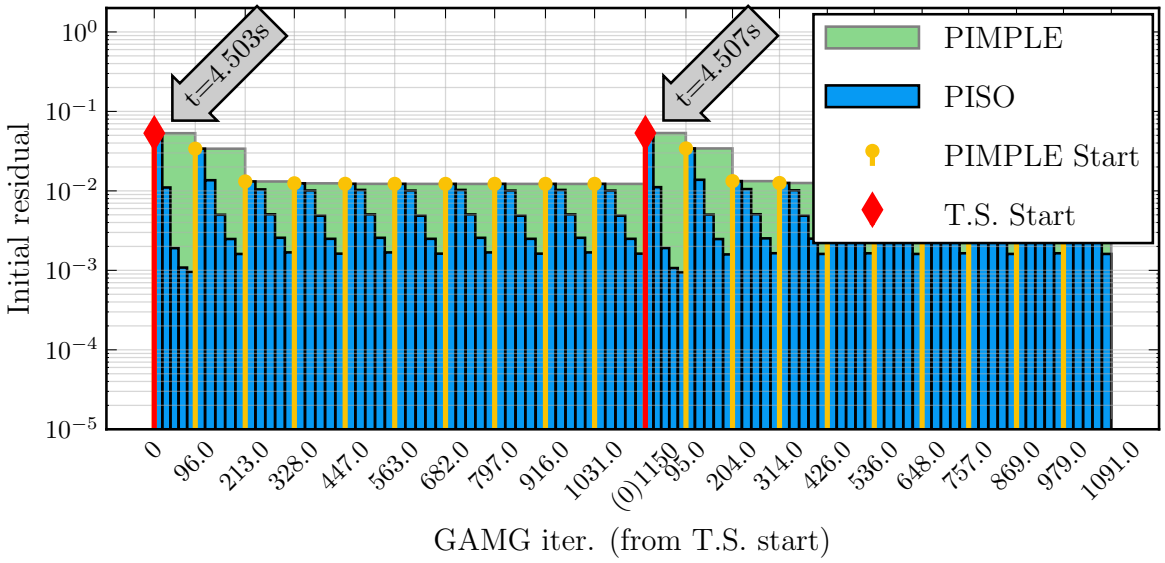


Figure 8.2 – Initial residual of the pressure equation and linear solver iteration distribution along two time steps; KCS in head regular wave; Fr=0.26; 10 PIMPLE iter.; 5 PISO iter.

8.2.1 Existing PIMPLE residual controllers

In order to control the number of PIMPLE iterations and therefore reduce the computational cost, several methods exist. A first one, available in OpenFOAM, consists in defining an absolute threshold value $\rho_{\text{err}}^{\text{PIM}}$ for the initial residual of a given field ϕ such that if the initial residual $r_{\phi}^{(\text{n,m,0})}$ at the beginning of a PIMPLE iteration is below this threshold, the current PIMPLE iteration is set as the final one for the current time step. With this method, it is possible to define a clear target value for the selected initial residual and to assume that the solution is converged when this value is reached. The problem of this method is that if the error saturates at an higher value than the expected one (as in Fig. 8.2), then the number of PIMPLE iterations is the user-defined maximal number of PIMPLE iterations.

Another method also available in OpenFOAM is the definition of a relative threshold value $\varepsilon_{\text{err}}^{\text{PIM}}$ such that, for the m$^{\text{th}}$ PIMPLE iteration in the n$^{\text{th}}$ time step, if $\frac{|r_{\phi}^{(\text{n,m,0})} - r_{\phi}^{(\text{n,1,0})}|}{r_{\phi}^{(\text{n,1,0})}} < \varepsilon_{\text{err}}^{\text{PIM}}$ then the current PIMPLE iteration is set as the final one for the current time step.

Finally, a third method implemented in *foamStar* consists in controlling the number of PIMPLE iterations with the initial residual $r_{\text{fsi}}^{(\text{n,m,0})}$ extracted from the body motion solver. If this residual is below a user-defined absolute threshold value $\rho_{\text{err}}^{\text{FSI}}$ then the current PIMPLE iteration is set as the final one for the current time step. With this method, the convergence of the body motion solver is well controlled. However some unwanted non-well solved fields far

from the body could not be captured by the body motion residual and then affect the global convergence of the simulation. The other default of this PIMPLE controller is that it can only be used considering moving body simulations.

8.2.2 Proposition of a new PIMPLE controller

In order to improve the control of the number of PIMPLE iterations, another PIMPLE controller is proposed in this section. The idea is to identify the saturation of the initial residual of the pressure by calculating the relative residual between two successive PIMPLE iterations as follows:

$$\varepsilon^{(\mathrm{n,m})} = \frac{|r_p^{(\mathrm{n,m,0})} - r_p^{(\mathrm{n,m\text{-}1,0})}|}{r_p^{(\mathrm{n,m,0})}} \tag{8.1}$$

The current PIMPLE iteration is set as the final one for the current time step if $\varepsilon^{(\mathrm{n,m})} < \varepsilon_{\mathrm{err}}^{\mathrm{SPIM}}$ where $\varepsilon_{\mathrm{err}}^{\mathrm{SPIM}}$ is a user-defined relative threshold value [1]. With such a PIMPLE controller, any stagnation of the pressure initial residual between two successive PIMPLE iterations implies the end of the PIMPLE loop, so no "unnecessary" PIMPLE iterations are run. The potential issue is that a transient stagnation of the initial residual even at an high value causes the stop of the PIMPLE loop while some additional iterations might have led to a better convergence.

Figure 8.3 shows the initial residuals of the pressure equation at the same time step than in Fig. 8.2 but using the new PIMPLE controller with $\varepsilon_{\mathrm{err}}^{\mathrm{SPIM}} = 0.1$. As expected, using this relative tolerance the number of PIMPLE iterations is reduced compared to what observed in Fig. 8.2. The number of GAMG iterations and the initial residuals between the uncontrolled simulation and the controlled one are really close one to each other for the first PIMPLE iterations they have in common.

1. SPIM exponent stands for "Successive PIMPLE".

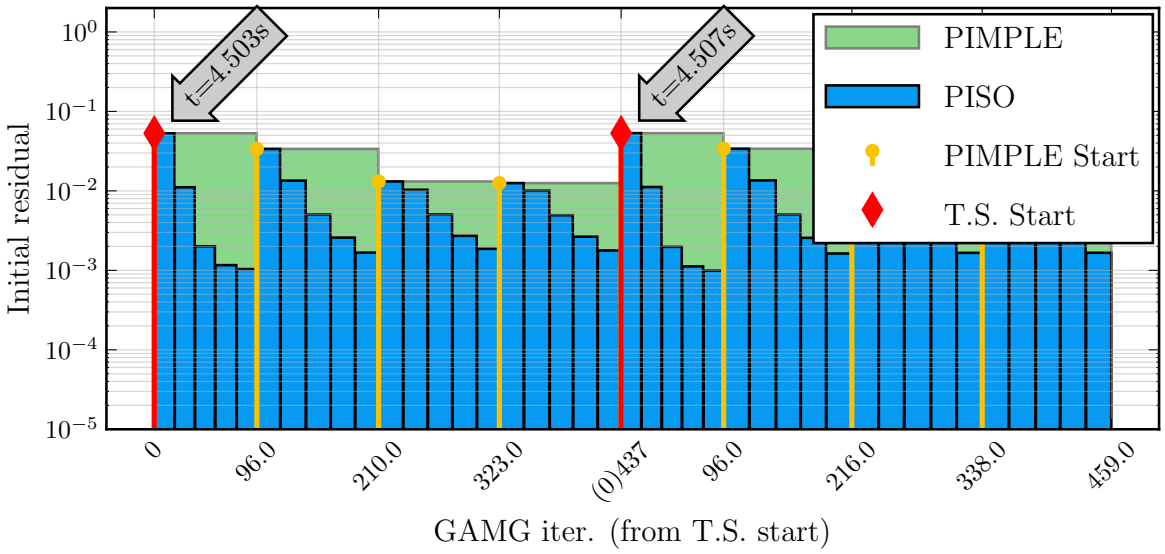


Figure 8.3 – Initial residual of the pressure equation and linear solver iteration distribution along two time steps; KCS in head regular wave; Fr=0.26; $\varepsilon_{\text{err}}^{\text{SPIM}} = 0.1$; 5 PISO iter.

Figure 8.4 shows the initial residuals of the pressure equation obtained by using the new PIMPLE controller with $\varepsilon_{\text{err}}^{\text{SPIM}} = 1$. Comparing to Fig. 8.3, this high relative tolerance reduces at two the number of PIMPLE iterations. The convergence of the initial residual between two successive PIMPLE iterations is no longer visible and the criterium $\varepsilon_{\text{err}}^{\text{SPIM}} = 1$ is potentially a too high value.

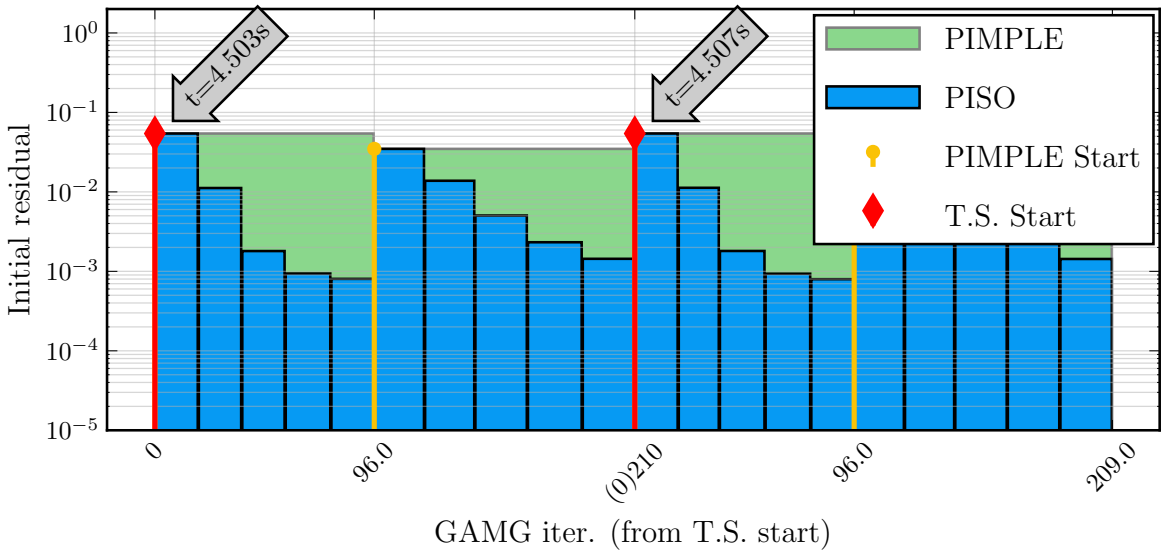


Figure 8.4 – Initial residual of the pressure equation and linear solver iteration distribution along two time steps; KCS in head regular wave; Fr=0.26; $\varepsilon_{\text{err}}^{\text{SPIM}} = 1$; 5 PISO iter.

An alternative is also tried, the condition $\varepsilon_{\text{err}}^{\text{SPIM}} = 1$ is kept but adding a body motion control with $\rho_{\text{err}}^{\text{FSI}} = 0.001$. Doing this, the PIMPLE loop is stopped at the m$^{\text{th}}$ iteration only if the two following conditions are verified:

$$r_{\text{fsi}}^{(\text{n,m,0})} < \rho_{\text{err}}^{\text{FSI}}$$
$$\text{and} \quad \frac{|r_p^{(\text{n,m,0})} - r_p^{(\text{n,m-1,0})}|}{r_p^{(\text{n,m,0})}} < \varepsilon_{\text{err}}^{\text{SPIM}}$$

Figure 8.5 shows the initial residuals of the pressure equation for such criteria. Comparing to Fig. 8.4, the number of PIMPLE iterations reaches three. So, in this case, the body motion residual tolerance is more restrictive than the newly implemented residual control [2].

2. With the present simulation this phenomenon was expected because a specific behavior is implemented for the body motion resolution. In fact no body motions are calculated at the second PIMPLE iteration in order to stabilize the computation. So, only using two PIMPLE iterations implies a single body motion resolution at the beginning of the time step. Consequently, with only two PIMPLE iterations, the body motion residual tolerance cannot be satisfied.

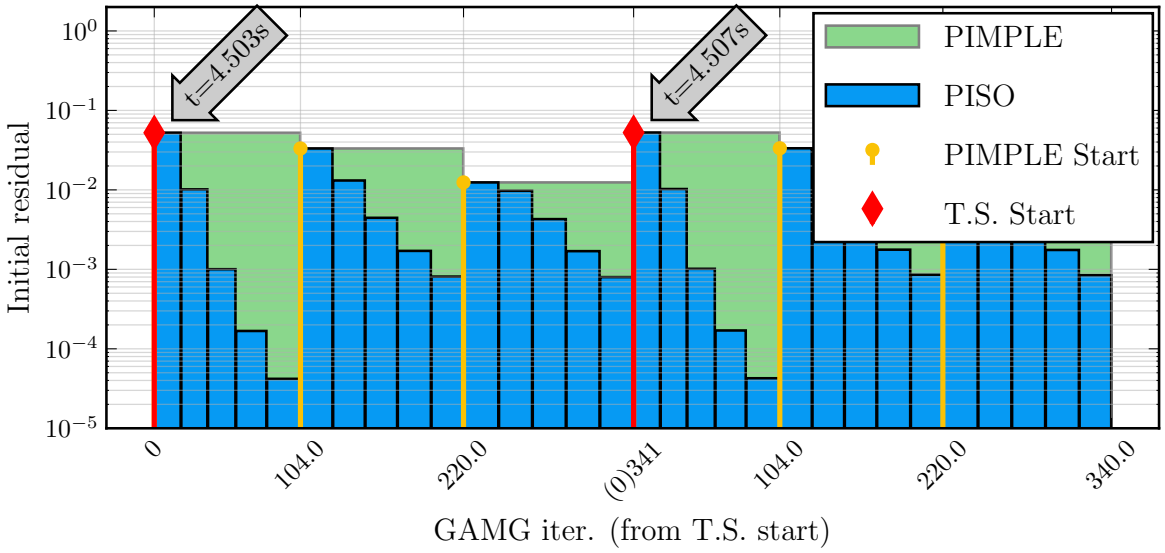


Figure 8.5 – Initial residual of the pressure equation and linear solver iteration distribution along two time steps; KCS in head regular wave; Fr=0.26; $\varepsilon_{\text{err}}^{\text{SPIM}} = 1$; $\rho\text{err}^{\text{FSI}} = 0.001$; 5 PISO iter.

For all the presented simulations, the simulated time interval is from $t^0 = 0$ s to $t^n = 5$ s. Table 8.1 synthesizes the computational cost of each simulation. As expected the computational cost is approximately reduced proportionally to the number of PIMPLE iterations avoided.

Table 8.1 – Computational cost of simulations using various PIMPLE controller methods and with a maximum of 10 PIMPLE iterations per time step.

	No control	$\varepsilon_{\text{err}}^{\text{SPIM}} = 0.1$	$\varepsilon_{\text{err}}^{\text{SPIM}} = 1$	$\varepsilon_{\text{err}}^{\text{SPIM}} = 1$ and $\rho_{\text{err}}^{\text{FSI}} = 0.001$
CPU time $(\times 10^4)$ [s]	5.512	2.254	1.086	1.882

Finally, it is now needed to assess the effect of reducing the number of PIMPLE iterations on the simulation results. For this simulation, the quantities of interest are the longitudinal resistance and the heave and pitch motions. Fig. 8.6 shows the time traces of the longitudinal pressure and viscous resistance coefficient when Fig. 8.7 shows the time traces of non-dimensionalized heave and pitch motions. Among all the curves only the method using $\varepsilon_{\text{err}}^{\text{SPIM}} = 1$ without defining $\rho_{\text{err}}^{\text{FSI}}$ differs slightly from the others during the time window simulated. Some non-well converged PIMPLE loops may be the cause of this phenomenon and the criterion $\varepsilon_{\text{err}}^{\text{SPIM}} = 1$ is probably a too high value to select. However, adding the second

criterion on $\rho_{\text{err}}^{\text{FSI}}$ seems to solve this issue.

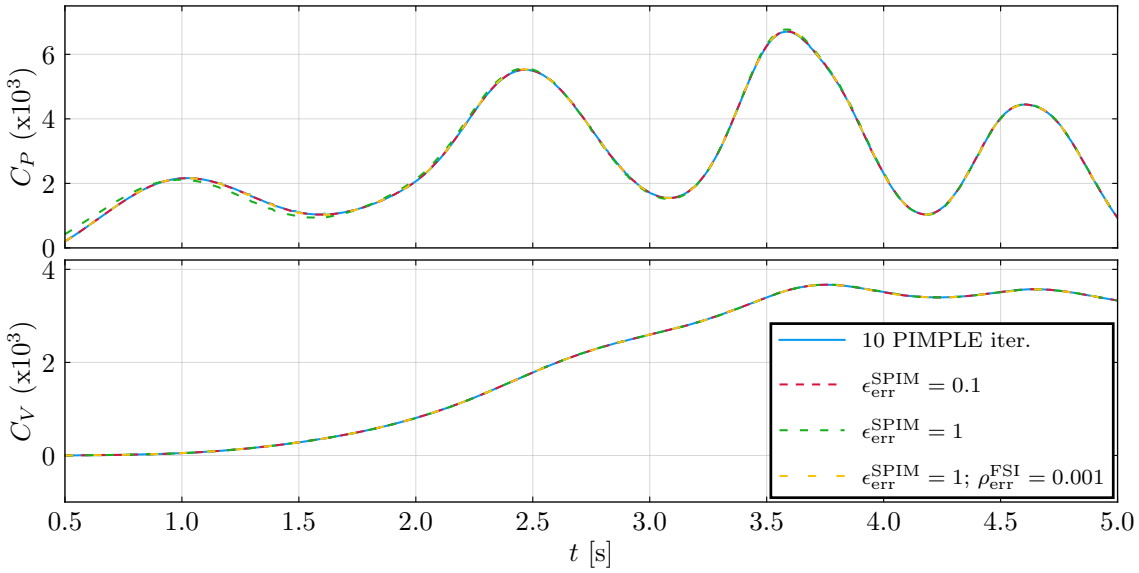


Figure 8.6 – Time history of the pressure resistance and viscous resistance coefficients using various PIMPLE controllers; KCS Fr = 0.26; regular head wave $\lambda = 0.65L_{PP}$; steepness $\varepsilon = 1.6\%$.

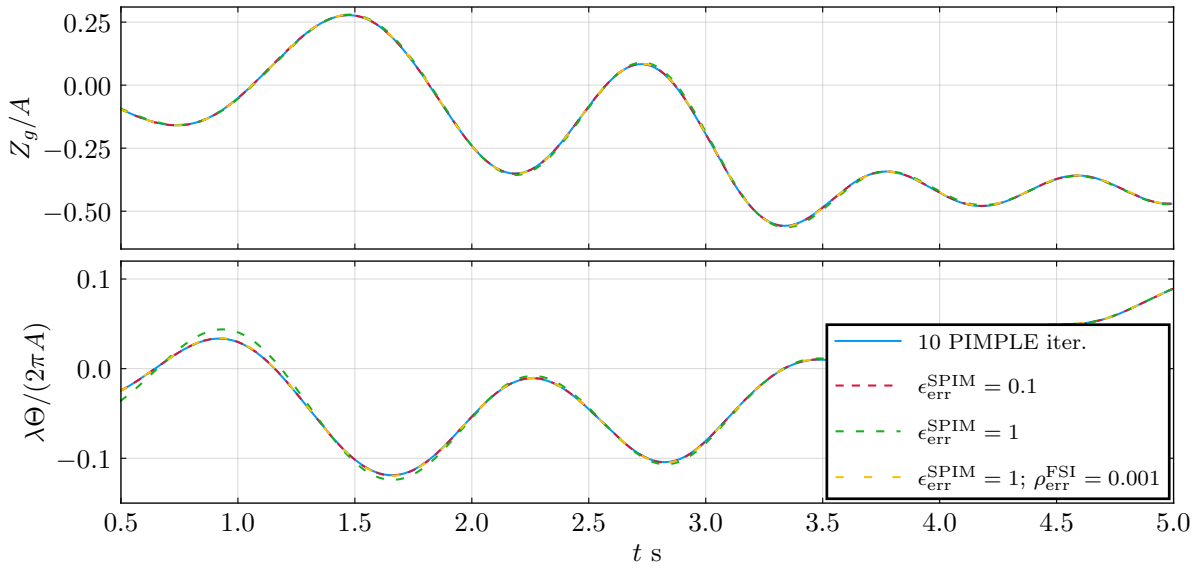


Figure 8.7 – Time history of the heave and pitch using various PIMPLE controllers; KCS Fr = 0.26; regular head wave $\lambda = 0.65L_{PP}$; steepness $\varepsilon = 1.6\%$

For this test case, the use of a body motion control $\rho_{\text{err}}^{\text{FSI}}$ has shown the best efficiency among all the tested methods. As mentioned, this PIMPLE controller is only applicable to simulations with a body motion and may not take into account some non-converged fields far from the body. The newly implemented PIMPLE controller has also shown good results. Keeping in mind the objective of efficiency, a wider range of test cases and more precise convergence analyses have to be done in order to define robust recommendations.

CHAPTER 9

CONCLUSION

9.1 Overview of the present work

The objectives of this PhD were to study and improve both the accuracy and efficiency of the in-house solver *foamStar*. The targeted applications were the computation of added wave resistance and ship motions in waves. To do so, work has been done looking at the code implementation, describing it in detail and doing some modifications intending to ease second-order accurate naval simulations. Besides, a progressive analysis was carried out, starting from mono-fluid Navier-Stokes simulations up to two-phase RANS simulations, in order to identify best practices for naval simulations with waves. Finally, some preliminary work was done qualifying the ability of *foamStar* and *foamStar-SWENSE* to run naval simulations with head waves.

9.1.1 Analysis of the numerical implementation

In the first part of this document a description of the implementation of *foamStar* is done. Chapter 2 presents the mathematical model used into *foamStar* and *foamStar-SWENSE* and then Chapter 3 gives a detailed description of the numerical implementation. Descriptions of the algorithms of *foamStar* were already done in Li et al. (2021), Choi et al. (2018) and Kim (2021). However, the objective during this Ph.D. was to describe as precisely as possible the numerical implementation and to build a complete flow-chart. The interest of such a flow-chart is to summarize the main algorithm of *foamStar* identifying each step of the numerical resolution and connecting them to the numerical equations.

After these investigations, some new implementations have been made. One of the key features of *foamStar* is the Multidimensional Universal Limiter for Explicit Solution (MULES) that is used solving the convective equation of the volume fraction with the VOF model. This method directly comes from the solver *interDymFoam* taken in its OpenFOAM-5 version. Initially, using the MULES algorithm only the first-order implicit Euler or a Crank-Nicolson

time scheme could be used. Some modifications were done in order to use the second-order backward time scheme with MULES.

9.1.2 Accuracy simulating nonlinear regular waves

The second part of this document presents a progressive analysis of the accuracy of *foamStar*. The objective was to propose a reference numerical set-up and mesh generation method targeting simulations of ship in waves with forward speed. A large part of this work is the direct continuation of studies done by Kim Y.J. (Kim, 2021; Kim et al., 2022a).

The reference numerical set-up used for this work was based on some recommendations made by previous works (Seng, 2012; Li, 2018; Choi et al., 2018; Kim, 2021) and the OpenFOAM literature. The time scheme was the main parameter to be studied comparing the second-order backward and the Crank-Nicolson time schemes. Verifying the spatial convergence, some specific mesh structures were also studied in order to quantify the impact of some mesh deformations and make recommendations for wave simulations.

The first study-case described was the Taylor-Green vortices. The objective was to verify the time schemes accuracy on mono-fluid Navier-Stokes simulations and then to identify the influence of cell aspect ratio, non-orthogonality and skewness on the solver accuracy. This study showed that for such simulations both the second-order backward and Crank-Nicolson schemes verify the second-order accuracy. Moreover, they both produced similar errors. Besides, the study showed that both the cell aspect ratio and non-orthogonality are accurately managed by *foamStar* for values remaining in an acceptable range from the OpenFOAM literature. However, with the studied cases, the cell skewness impaired significantly the accuracy. It was shown that the use of a skewness corrector implemented in OpenFOAM and usable in *foamStar* could be a convenient solution.

The next study-case was the nonlinear regular wave propagation in a periodic bi-dimensional domain of one wavelength. Only the wavelength $\lambda = 1$ m was considered but three wave steepnesses were studied: 1%, 5% and 10%. This study raised four points. First, for simulations of regular waves using the Navier-Stokes equations, a converged air velocity field can be identified and it is far from the null velocity profile commonly imposed using *foamStar*. Besides, using MULES, the accuracy of both the newly implemented second-order backward and Crank-Nicolson schemes is consistent with what would be expected from a second-order scheme. Then, some gain of energy with time was identified. The backward scheme tended to show a larger sensibility to this phenomenon. An assumption was that the selected numerical set-up and its use with *foamStar* leads to non-conservative simulations in terms of energy. Some

future work on the energy conservation in *foamStar* could answer this potential issue. Finally, a important part of Chapt. 5 was dedicated to the comparison of the numerical errors obtained using different mesh refinements and cell aspect ratio at the free surface. With the studied mesh generation method, an optimal range of free-surface cell aspect ratios was defined. This optimal range depends on the wave steepness.

The last study-case of this part was the nonlinear regular wave propagation in a long bi-dimensional domain of several wavelengths. The same waves as for the periodic domain were studied. The objectives were to verify the accuracy of both the second-order backward and Crank-Nicolson schemes and to control if the mesh refinement recommendations made for periodic domains remain convenient with a long simulation domain, relaxation zones, RANS model and a moving reference frame. The results showed a good agreement between the error of both the backward and Crank-Nicolson schemes and the expected second-order accuracy. Besides, the mesh recommendations remained valid with the tested cases. However, adding relaxation zones generated perturbations in the air. The reason is that to impose a null air velocity as a targeted value leads to the spurious vortices generation in the air. Removing relaxation zone at the inlet of the domain gave some improvements of the results. This modification had to be discussed regarding to the need of damping some diffracted field for simulations with bodies. Finally, using a moving reference frame also changed the accuracy of *foamStar* and, clear numerical convergence was no more identified.

9.1.3 Naval applications

During this Ph.D. a preliminary study assessed the ability of *foamStar* and *foamStar-SWENSE* to manage simulation of a ship moving with forward speed in head regular waves. The study was based on the Tokyo 2015 workshop simulating a scale model of a KRISO container ship with forward speed and five different regular head wave conditions. *foamStar* and *foamStar-SWENSE* were compared but no mesh convergence and efficiency analyses were done. The result were in good agreement with the experiments but a complete validation procedure remains to do. A simulation of ship in long-crested irregular head waves was also performed, however, no validation or comparison with other numerical results were done. These studies were carried out using the first-order Euler time scheme because of stability reasons. When these studies were conducted, the second-order backward MULES were not implemented and some recommendations concerning the numerical set up and the mesh structure were not established yet. One of the test cases initially simulated was thus simulated again using those backward scheme and configuration recommendations. They provided clear enhancement of the previous

results. This was the last work of the thesis and more detailed and exhaustive future work have to be done in order to validate *foamStar* using the recommendations made all along this Ph.D.

The last chapter of this document gives a synthesis of the work done in order to integrate *foamStar* into the industrial process of BVS-M&O. Some user-guides are tutorial were made and a scripts was implemented in order to help the preparation of numerical set up following the selected recommendations.

9.2 Perspectives and proposal for future works

This Ph.D. took place in the context of a research Chair. established by BV and École Central Nantes in partnership with BVS-M&O. Consequently, the work presented was in the continuity of the work done by the other members of the Chair. also working on the development of *foamStar*. The following presents some future work proposal on the improvement of the *foamStar* accuracy/efficiency and its use for naval simulation in an industrial context.

In the direct continuity of the Ph.D., validations of the selected numerical set-up with naval simulation in head and oblique regular waves have to be done. Some work on the efficiency not presented in this document also has to be continued in order to optimize the numerical set-up reducing the computational costs. For instance, some directions are: optimizing the number of PIMPLE and PISO iterations, adding or modifying some numerical correctors or optimizing the choice of the linear solvers used for the matrix resolutions.

Another aspect that was not studied deeply during this Ph.D. is the optimization and improvement of *foamStar-SWENSE*. Because of some promising results shown in Li et al. (2021) and Kim (2021) some future work should be done on this topic.

Besides, the current implementation of the relaxation zones has shown some problematic aspects and alternative implementations of the relaxation zone should be studied with possibly some other form of field extrapolation in the air for the targeted wave. Some extrapolation techniques avoiding a null air velocity were already proposed in Li (2018) and Choi (2019).

Finally, an important objective for the future of *foamStar* is to simulate ship moving in complex irregular sea states. The work presented in Kim (2021) already assesses the ability of *foamStar* to simulate such sea states without ship. The next step is to complete the preliminary trials shown in this document carrying out some studies on the ship response with simulations of long and short-crested irregular waves from various directions.

BIBLIOGRAPHY

S. Aliyar, G. Ducrozet, B. Bouscasse, S. Venkatachalam, and P. Ferrant. Breaking focused wave interaction with cylinder using hos-openfoam coupling. In *OCEANS 2022 - Chennai*, pages 1–10, 2022. doi: 10.1109/OCEANSChennai45887.2022.9775539.

X. Barthelemy, M. Banner, W. Peirson, F. Fedele, M. Allis, and F. Dias. On a unified breaking onset threshold for gravity waves in deep and intermediate depth water. *Journal of Fluid Mechanics*, 841:463–488, 2018.

A. Benhamou, S. Seng, C. Monroy, J. de Lauzon, and S. Malenica. Hydroelastic simulations in openfoam®: A case study on a 4400teu containership. In *8th Int. Conf. on Hydroelasticity in Marine Technology. SNU, Seoul, Korea*, 2018.

B. Bouscasse, A. Califano, Y. M. Choi, X. Haihua, J. W. Kim, Y. J. Kim, S. H. Lee, H.-J. Lim, D. M. Park, M. Peric, et al. Qualification criteria and the verification of numerical waves: Part 2: Cfd-based numerical wave tank. In *International Conference on Offshore Mechanics and Arctic Engineering*, volume 85116, page V001T01A007. American Society of Mechanical Engineers, 2021.

I. Celik. Numerical uncertainty in fluid flow calculations: needs for future research. 1993.

Y. Choi. *Two-way coupling between potential and viscous flows for a marine application*. PhD thesis, École centrale de Nantes, 2019.

Y. Choi, B. Bouscasse, S. Seng, G. Ducrozet, L. Gentaz, and P. Ferrant. Generation of regular and irregular waves in Navier-Stokes CFD solvers by matching with the nonlinear potential wave solution at the boundaries. In *37th International Conference on Ocean, Offshore and Arctic Engineering*. American Society of Mechanical Engineers, 2018.

Y.-M. Choi, Y. J. Kim, B. Bouscasse, S. Seng, L. Gentaz, and P. Ferrant. Performance of different techniques of generation and absorption of free-surface waves in computational fluid dynamics. *Ocean Engineering*, 214:107575, 2020. ISSN 0029-8018. doi: https://doi.org/10.1016/j.oceaneng.2020.107575. URL `https://www.sciencedirect.com/science/article/pii/S0029801820305837`.

H. W. Coleman and F. Stern. Uncertainties and cfd code validation. 1997.

M. Darwish and F. Moukalled. Tvd schemes for unstructured grids. *International Journal of heat and mass transfer*, 46(4):599–611, 2003.

M. Darwish and F. Moukalled. *The finite volume method in computational fluid dynamics: an advanced introduction with OpenFOAM® and Matlab®*. Springer, 2021.

D. G. Dommermuth and D. K. Yue. A high-order spectral method for the study of nonlinear gravity waves. *Journal of Fluid Mechanics*, 184:267–288, 1987.

G. Ducrozet, F. Bonnefoy, D. Le Touzé, and P. Ferrant. A modified high-order spectral method for wavemaker modeling in a numerical wave tank. *European Journal of Mechanics-B/Fluids*, 34:19–34, 2012.

G. Ducrozet, F. Bonnefoy, D. Le Touzé, and P. Ferrant. HOS-ocean: Open-source solver for nonlinear waves in open ocean based on High-Order Spectral method. *Computer Physics Communications*, 203:245–254, 2016.

G. Ducrozet, B. Bouscasse, M. Gouin, P. Ferrant, and F. Bonnefoy. CN-Stream: Open-source library for nonlinear regular waves using stream function theory. *arXiv:1901.10577 [physics.flu-dyn]*, pages 1–31, 2019.

L. Eça and M. Hoekstra. Verification and validation for marine applications of cfd. *International shipbuilding progress*, 60(1-4):107–141, 2013.

L. Eça and M. Hoekstra. A procedure for the estimation of the numerical uncertainty of CFD calculations based on grid refinement studies. *Journal of Computational Physics*, 262: 104–130, 2014. ISSN 10902716. doi: 10.1016/j.jcp.2014.01.006.

F. N. Felten and T. S. Lund. Kinetic energy conservation issues associated with the collocated mesh scheme for incompressible flow. *Journal of Computational Physics*, 215(2):465–484, 2006.

Y. Feng, O. el Moctar, and T. E. Schellin. Parametric hull form optimization of containerships for minimum resistance in calm water and in waves. *Journal of Marine Science and Application*, pages 1–24, 2022.

P. Ferrant, L. Gentaz, B. Alessandrini, and D. Le Touzé. A potential/RANSE approach for regular water wave diffraction about 2-D structures. *Ship Technology Research*, 50(4): 165–171, 2003.

J. H. Ferziger and M. Peric. *Computational methods for fluid dynamics*. Springer Science & Business Media, 2012.

foam-extend 3.1. foam-extend-3.1: Installation guide. Technical report, 2004. URL `https://openfoamwiki.net/index.php/Installation/Linux/foam-extend-3.1`.

J. S. Forrest and I. Owen. An investigation of ship airwakes using detached-eddy simulation. *Computers & Fluids*, 39(4):656–673, 2010.

H. Fujii. Experimental study on the resistance increase of a ship in regular oblique waves. *Proc. of 14th ITTC, 1975*, 4:351–360, 1975.

P. Gaskell and A. Lau. Curvature-compensated convective transport: Smart, a new boundedness-preserving transport algorithm. *International Journal for numerical methods in fluids*, 8(6):617–641, 1988.

C. Greenshields and H. Weller. *Notes on Computational Fluid Dynamics: General Principles*. CFD Direct Ltd, Reading, UK, 2022.

A. Harten. On a class of high resolution total-variation-stable finite-difference schemes. *SIAM Journal on Numerical Analysis*, 21(1):1–23, 1984.

R. G. Hills. Model validation: model parameter and measurement uncertainty. 2006.

T. Hino. Proceedings of cfd workshop tokyo 2005. *Tokyo, Japan*, 2005.

C. W. Hirt and B. D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39(1):201–225, 1981. ISSN 10902716. doi: 10.1016/0021-9991(81)90145-5.

O. Hizir, M. Kim, O. Turan, A. Day, A. Incecik, and Y. Lee. Numerical studies on non-linearity of added resistance and ship motions of kvlcc2 in short and long waves. *International Journal of Naval Architecture and Ocean Engineering*, 11(1):143–153, 2019.

J. Hou, F. Simons, and R. Hinkelmann. Improved total variation diminishing schemes for advection simulation on arbitrary grids. *International Journal for Numerical Methods in Fluids*, 70(3):359–382, 2012.

Z. Z. Hu, D. Greaves, and A. Raby. Numerical wave tank study of extreme waves and wave-structure interaction using openfoam®. *Ocean Engineering*, 126:329–342, 2016.

N. G. Jacobsen, D. R. Fuhrman, and J. Fredsøe. A wave generation toolbox for the open-source CFD library: OpenFoam®. *International Journal for Numerical Methods in Fluids*, 70(9): 1073–1088, 2012. ISSN 02712091. doi: 10.1002/fld.2726.

H. Jasak. Error analysis and estimation for the finite volume method with applications to fluid flows. 1996.

H. Jasak, A. Jemcov, and J. P. Maruszewski. Preconditioned linear solvers for large eddy simulation. In *CFD 2007 Conference, CFD Society of Canada*, 2007a.

H. Jasak, A. Jemcov, Z. Tukovic, et al. Openfoam: A c++ library for complex physics simulations. In *International workshop on coupled methods in numerical dynamics*, volume 1000, pages 1–20. IUC Dubrovnik Croatia, 2007b.

A. Jemcov, J. P. Maruszewski, and H. Jasak. Performance improvement of algebraic multigrid solver by vector sequence extrapolation. In *CFD 2007 Conference, CFD Society of Canada*. Citeseer, 2007.

S. Kianejad, H. Enshaei, J. Duffy, and N. Ansarifard. Prediction of a ship roll added mass moment of inertia using numerical simulation. *Ocean Engineering*, 173:77–89, 2019.

M. Kim, O. Hizir, O. Turan, S. Day, and A. Incecik. Estimation of added resistance and ship speed loss in a seaway. *Ocean Engineering*, 141:465–476, 2017.

W. Kim, D. Van, and D. Kim. Measurement of flows around modern commercial ship models. *Exp. in Fluids*, 31:567–578, 2001.

Y.-J. Kim. *Numerical improvement and validation of a naval hydrodynamics CFD solver in view of performing fast and accurate simulation of complex ship-wave interaction*. PhD thesis, École Centrale de Nantes, 2021.

Y. J. Kim, B. Bouscasse, S. Seng, and D. Le Touzé. Efficiency of diagonally implicit runge-kutta time integration schemes in incompressible two-phase flow simulations. *Computer Physics Communications*, page 108415, 2022a.

Y.-R. Kim, E. Esmailian, and S. Steen. A meta-model for added resistance in waves. *Ocean Engineering*, 266:112749, 2022b.

Y. Kodama, H. Takeshi, M. Hinatsu, T. Hino, S. Uto, N. Hirata, and S. Murashige. Proceedings of the 1994 cfd workshop. *Ship Research Institute, Japan*, 1994.

B. E. Larsen and D. R. Fuhrman. On the over-production of turbulence beneath surface waves in reynolds-averaged navier–stokes models. *Journal of Fluid Mechanics*, 853:419–460, 2018.

B. E. Larsen, D. R. Fuhrman, and J. Roenby. Performance of interFoam on the simulation of progressive waves. *Coastal Engineering Journal*, 61(3):380–400, 2019. ISSN 17936292. doi: 10.1080/21664250.2019.1609713.

L. Larsson, F. Stern, and M. Visonneau. Gothenburg 2010, a workshop on numerical ship hydrodynamics. Technical report, Chalmers University of Technology, 2010.

L. Larsson, F. Stern, and M. Visonneau. *Numerical ship hydrodynamics: an assessment of the Gothenburg 2010 workshop*. Springer, 2013.

L. Larsson, F. Stern, M. Visonneau, T. Hino, N. Hirata, and J. Kim. Tokyo 2015: A workshop on CFD in ship hydrodynamics. In *Workshop Proceedings, Tokyo, Dec*, 2018.

Q. Li, J. Wang, S. Yan, J. Gong, and Q. Ma. A zonal hybrid approach coupling fnpt with openfoam for modelling wave-structure interactions with action of current. *Ocean Systems Engineering*, 8(4):381–407, 2018a.

Z. Li. *Two-phase spectral wave explicit Navier-Stokes equations method for wave-structure interactions*. PhD thesis, École centrale de Nantes, 2018.

Z. Li, B. Bouscasse, L. Gentaz, G. Ducrozet, and P. Ferrant. Progress in coupling potential wave models and two-phase solvers with the SWENSE methodology. In *37th International Conference on Ocean, Offshore and Arctic Engineering*. American Society of Mechanical Engineers, 2018b.

Z. Li, B. Bouscasse, G. Ducrozet, L. Gentaz, D. Le Touzé, and P. Ferrant. Spectral Wave Explicit Navier-Stokes Equations for wave-structure interactions using two-phase Computational Fluid Dynamics solvers. *Ocean Engineering*, 221:108513, 2021. ISSN 0029-8018. doi: https://doi.org/10.1016/j.oceaneng.2020.108513. URL `https://www.sciencedirect.com/science/article/pii/S0029801820314207`.

R. Luquet, G. Ducrozet, L. Gentaz, P. Ferrant, and B. Alessandrini. Applications of the swense method to seakeeping simulations in irregular waves. In *Proc of the 9th Int. Conf. on Num. Ship Hydro*, 2007.

S. Márquez Damián. An extended mixture model for the simultaneous treatment of short and long scale interfaces. 2013.

H. Marschall. *Towards the numerical simulation of multi-scale two-phase flows*. PhD thesis, Technische Universität München, 2011.

M. McHale, J. Friedman, and J. Karian. Standard for verification and validation in computational fluid dynamics and heat transfer. *The American Society of Mechanical Engineers, ASME V&V*, 20, 2009.

F. R. Menter. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA journal*, 32(8):1598–1605, 1994.

A. Mofidi, J. E. Martin, and P. M. Carrica. Propeller/rudder interaction with direct and coupled cfd/potential flow propeller approaches, and application to a zigzag manoeuvre. *Ship technology research*, 65(1):10–31, 2018.

C. Monroy, G. Ducrozet, F. Bonnefoy, A. Babarit, L. Gentaz, and P. Ferrant. RANS simulations of CALM buoy in regular and irregular seas using SWENSE method. *International Journal of Offshore and Polar Engineering*, 21(4):264–271, 2011. ISSN 10535381.

OpenFOAM. Openfoam: User guide. Technical report, 2022a. URL `https://cfd.direct/openfoam/user-guide/`.

OpenFOAM. Openfoam: History. Technical report, 2022b. URL `https://www.openfoam.com/news/history`.

OpenFOAM. Openfoam: User guide. Technical report, 2022c. URL `https://www.openfoam.com/documentation/user-guide`.

F. Pereira, L. Eça, and G. Vaz. Verification and validation exercises for the flow around the kvlcc2 tanker at model and full-scale reynolds numbers. *Ocean Engineering*, 129:133–148, 2017.

G. Reliquet, A. Drouet, P. Guillerm, L. Gentaz, and P. Ferrant. Simulation of wave-ship interaction in regular and irregular seas under viscous flow theory using the swense method. In *Proc. 30th Symp. Naval Hydrod.(Tasmania, 11/2014)*, 2014.

C. Rhie and W. L. Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA journal*, 21(11):1525–1532, 1983.

M. M. Rienecker and J. D. Fenton. A Fourier approximation method for steady water waves. *Journal of Fluid Mechanics*, 104:119–137, 1981. doi: 10.1017/S0022112081002851.

P. J. Roache. *Verification and validation in computational science and engineering*, volume 895. Hermosa Albuquerque, NM, 1998.

P. J. Roache. Validation in fluid dynamics and related fields. In *Computer Simulation Validation*, pages 661–683. Springer, 2019.

H. Rusche. *Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions*. PhD thesis, Imperial College of Science, Technology and Medicine, 2002. URL `http://portal.acm.org/citation.cfm?doid=1806799.1806850`.

H. Rusche. *Computational fluid dynamics of dispersed two-phase flows at high phase fractions*. PhD thesis, Imperial College London (University of London), 2003.

B. R. Seiffert, G. Ducrozet, and F. Bonnefoy. Simulation of breaking waves using the high-order spectral method with laboratory experiments: Wave-breaking onset. *Ocean Modelling*, 119: 94–104, 2017.

S. Seng. *Slamming and Whipping Analysis of Ships*. PhD thesis, Technical University of Denmark, 2012.

S. Seng, J. J. Jensen, and Š. Malenica. Global hydroelastic model for springing and whipping based on a free-surface CFD code (OpenFOAM). *International Journal of Naval Architecture and Ocean Engineering*, 6(4):1024–1040, 2014. ISSN 20926790. doi: 10.2478/IJNAOE-2013-0229.

S. Seng, C. Monroy, and S. Malenica. On the use of euler and crank-nicolson time-stepping schemes for seakeeping simulations in openfoam. In *MARINE VII: proceedings of the VII International Conference on Computational Methods in Marine Engineering*, pages 905–920. CIMNE, 2017.

A. Serani, F. Stern, E. F. Campana, and M. Diez. Hull-form stochastic optimization via computational-cost reduction methods. *Engineering with Computers*, pages 1–25, 2021.

S. Sigmund and I. M. Peric. *Performance of Ships in Waves*. PhD thesis, Universität Duisburg-Essen, 2018.

C. D. Simonsen, J. F. Otzen, S. Joncquez, and F. Stern. Efd and cfd for kcs heaving and pitching in regular head waves. *Journal of Marine Science and Technology*, 18(4):435–459, 2013.

R. Skejic and O. M. Faltinsen. A unified seakeeping and maneuvering analysis of ships in regular waves. *Journal of marine science and technology*, 13(4):371–394, 2008.

P. R. Spalart. Detached-eddy simulation. *Annual review of fluid mechanics*, 41:181–202, 2009.

F. Stern, M. Diez, H. Sadat-Hosseini, H. Yoon, and F. Quadvlieg. Statistical approach for computational fluid dynamics state-of-the-art assessment: N-version verification and validation. *Journal of Verification, Validation and Uncertainty Quantification*, 2(3), 2017.

R. Subramanian and R. F. Beck. A time-domain strip theory approach to maneuvering in a seaway. *Ocean Engineering*, 104:107–118, 2015.

P. K. Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM journal on numerical analysis*, 21(5):995–1011, 1984.

Z. Tian, M. Perlin, and W. Choi. An eddy viscosity model for two-dimensional breaking waves and its validation with laboratory experiments. *Physics of Fluids*, 24(3):036601, 2012.

S. Van, W. Kim, G. Yim, D. Kim, and C. Lee. Experimental investigation of the flow characteristics around practical hull forms. *Proceedings 3rd Osaka Colloquium on Advanced CFD Applications to Ship Flow and Hull Form Design, Osaka, Japan*, 1998.

H. A. Van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM Journal on scientific and Statistical Computing*, 13 (2):631–644, 1992.

V. Vukcevic. *Numerical Modelling of Coupled Potential and Viscous Flow for Marine Applications*. PhD thesis, University of Zagreb, 2016.

H. Weller and A. Shahrokhi. Curl-free pressure gradients over orography in a solution of the fully compressible euler equations with implicit treatment of acoustic and gravity waves. *Monthly Weather Review*, 142(12):4439–4457, 2014.

B. J. West, K. A. Brueckner, R. S. Janda, D. M. Milder, and R. L. Milton. A new numerical method for surface hydrodynamics. *Journal of Geophysical Research: Oceans*, 92(C11): 11803–11824, 1987.

W. Zhang and O. El Moctar. Numerical prediction of wave added resistance using a rankine panel method. *Ocean Engineering*, 178:66–79, 2019.

Titre : Analyse numérique et développement de modèles précis dans un solveur CFD dédié aux applications navales avec houle

Mot clés : OpenFOAM ; Schéma temporel backward du deuxième ordre ; Génération de vagues ; Résistance ajoutée sur houle ; Tenue à la mer du navire

Résumé : L'objectif de cette thèse est de développer des solveurs numériques et des méthodologies afin d'améliorer le temps de calcul et la précision des simulations de tenue à la mer et de résistance ajoutée sur houle.
Tout d'abord, une synthèse de l'algorithme du solver *foamStar* développé en interne est effectuée. A partir de cette analyse, une modification est proposée afin de pouvoir utiliser le "Multidimensional Universal Limiter for Explicit Solution" (MULES) avec un schéma temporel backward d'ordre deux.
Ensuite, plusieurs études successives sont réalisées afin de : vérifier l'implémentation du schéma backward ; définir une configuration numérique et des maillages efficaces pour la simulation de houle. Les cas d'étude sont : les tourbillons de Taylor-Green, la houle régulière non linéaire se propageant dans un domaine periodique, et enfin, la houle régulière générée avec des zones de relaxation et des configurations numériques proches de celles utilisées pour des applications navales.
Dans la dernière partie de cette thèse, une étude préliminaire est réalisée en simulant un porte-conteneur avec une vitesse d'avance dans des houles régulières de face. Les recommandations définies tout au long de cette thèse sont également évaluées.

Title: Numerical analysis and development of accurate models in a CFD solver dedicated to naval applications with waves

Keywords: OpenFOAM; Second-order backward time scheme; Regular wave generation; Wave added resistance; Ship seakeeping

Abstract: The objective of the present thesis is to develop solvers and methodologies in order to improve the computational cost and the accuracy with regard to the thematics of seakeeping and added resistance.
First, a synthetic workflow of the algorithm of the in-house solver *foamStar* is proposed. From this analysis a modification is proposed in order to use the Multidimensional Universal Limiter for Explicit Solution (MULES) with a second-order backward time scheme.
Then, successive studies are done in order to: verify the implementation of the backward scheme; define an efficient numerical set-up and adequate mesh structures for numerical wave simulations. The case studies are, Taylor-Green vortices, nonlinear regular wave propagating in a periodic domain, and finally, regular waves generated with relaxation zones considering numerical configurations close to what is used for naval applications.
In the last part of this Thesis, a preliminary study is done simulating a containership with forward speed in head regular waves. The recommendations derived all along this thesis are also evaluated.