

---

**Numerical Methods for Water Flows  
with Free-Surface Gravity Waves**

---

**Mervyn Lewis**

© Mervyn Rudy Lewis, Amsterdam 2004

The research reported in this thesis has been carried out at the Centre for Mathematics and Computer Science (CWI) Amsterdam and was supported by the Dutch Technology Foundation STW (under grant no. CWI.4883).

# Numerical Methods for Water Flows with Free-Surface Gravity Waves

Proefschrift

ter verkrijging van de graad van doctor  
aan de Technische Universiteit Delft,  
op gezag van de Rector Magnificus  
prof.dr.ir. J.T. Fokkema,  
voorzitter van het College voor Promoties,  
in het openbaar te verdedigen  
op dinsdag 8 juni 2004 om 10:30 uur

door

**Mervyn Rudy LEWIS**

*ingenieur Luchtvaart- en Ruimtevaarttechniek  
geboren te Zwolle*

Dit proefschrift is goedgekeurd door de promotor:  
prof.dr.ir. B. Koren

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof.dr.ir. B. Koren,	Technische Universiteit Delft / CWI, promotor
Prof.dr. P.W. Hemker,	Universiteit van Amsterdam / CWI
Prof.dr.ir. G.S. Stelling,	Technische Universiteit Delft
Prof.dr.ir. T.J.C. van Terwisga,	Technische Universiteit Delft / MARIN
Prof.dr.ir. P. Wesseling,	Technische Universiteit Delft
Dr.ir. E.H. van Brummelen,	Technische Universiteit Delft
Dr.ir. A. van der Ploeg,	MARIN



## Preface

This thesis contains a record of the research I have done as a member of the scientific cluster Modelling, Analysis and Simulation at the Centre for Mathematics and Computer Science (CWI) in Amsterdam, during the period September 1999 until November 2003.

This research was financially supported by the Dutch Technology Foundation STW. I would like to thank STW's program officers drs. Marijke de Jong and Dipl.Phys. Margriet Jansz for their constructive collaboration. Furthermore, I would like to extend my gratitude to the members of the 'STW-gebruikerscommissie': dr.ir. Johan Romate (Shell), dr.ir. Mart Borsboom (WL-Delft Hydraulics), and last but not least dr.ir. Hoyte Raven and dr.ir. Martin Hoekstra (both MARIN). My research has benefitted from their constant critical views towards it and from the many stimulating discussions.

I would also like to thank the members of the *Resistance and Propulsion Research* group at MARIN for providing me with satisfactory answers to my questions concerning PARNASSOS and for assisting me with the generation of the meshes for my numerical experiments. Their enthusiasm and willingness to share their expertise during our many fruitful discussions on free-surface flows throughout these last four years, particularly during my stay at MARIN, were very inspiring and are gratefully acknowledged.

Now on a more personal note. I would like to begin by stating that a (large) part of my job satisfaction can be contributed to the pleasant working environment at CWI. Accordingly, I would like to say "thank you" to all my CWI-colleagues, for creating this atmosphere. There are, however, a number of persons who have contributed significantly, in one form or another, to the completion of this thesis and, hence, cannot be left unrecognized.

First and foremost I am deeply indebted to professor Barry Koren, my thesis supervisor and co-worker. His critical reading of my drafts, including the one for this thesis, and the cooperation during the project were indispensable. I am very much in awe of his eye for detail and his expert knowledge of CFD, from which this thesis has benefitted immensely. Secondly, I wish to extend my gratitude to professor Piet Hemker. His clear and concise way of explaining has really improved my understanding of numerical analysis and mathematics in general. It was my fortune to have been supervised by these two excellent researchers. Furthermore, I would like to thank Harald van Brummelen. Sharing an office with him has led to many pleasant discussions concerning our mutual interests, i.e., free-surface flows

and shotokan karate. His assistance in understanding the former was advantageous (despite his sometimes rather unconventional mathematical notations). Miranda van Uitert and Debby Lanser, my fellow PhD students, deserve tremendous credit. Their support during this period was indispensable. Our (almost) daily tea breaks were a welcome distraction. Besides Harald, I had the pleasure of sharing my office with Menno Genseberger and later on with Domenico Lahaye and David Echeverria. I have enjoyed their company (and I hope they have enjoyed mine). If it was not for drs. Margreet Nool, it would have taken me a lot longer to fully understand (and appreciate) FORTRAN 95. Also, our visit to the AIAA-CFD conference in Orlando will not be easily forgotten. Thanks Margreet ! Moreover, I would like to thank Jason Frank, for our many scientific and not-so-scientific discussions during our visits to “Brouwerij 't IJ”, and Johannes Krottje, for our daily game of ‘ping-pong’.

In my spare time I enjoyed the company of my fellow *PhDays*-organizers, Martijn, Serge, Debby and Harald, resulting in the *PhDays 2000*, and later on Erwin, Willemijn and Caroline, resulting in the *PhDays 2001*. Finally, I would like to extend my gratitude to my ‘paranimfen’ Miranda van Uitert and Jenny Dirks, the members of karate club *Dokan*, my family, and Vesna. They provided the necessary and appropriate boundary conditions for the successful completion of this project.

Utrecht, February 2004

Mervyn Lewis.

# Contents

<b>Preface</b>	<b>i</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Application area . . . . .	1
1.2 Basic grid techniques . . . . .	3
1.2.1 Moving-grid techniques . . . . .	4
1.2.2 Fixed-grid techniques . . . . .	5
1.3 Outline . . . . .	8
1.3.1 Part A: fitting method . . . . .	8
1.3.2 Part B: capturing method . . . . .	9
<b>2 Efficient Computation of Steady, 3D Water-Wave Patterns</b>	<b>13</b>
2.1 Introduction . . . . .	13
2.2 Governing equations and free-surface conditions . . . . .	14
2.2.1 Flow equations . . . . .	14
2.2.2 Free-surface conditions . . . . .	14
2.2.3 Quasi free-surface boundary condition . . . . .	15
2.3 Computational method . . . . .	16
2.3.1 Free-surface iteration method . . . . .	16
2.3.2 Discrete Navier-Stokes boundary-value problem . . . . .	16
2.4 Numerical results . . . . .	19
2.4.1 First results . . . . .	19
2.4.2 Influence of outflow boundary condition . . . . .	22
2.4.3 Influence of preconditioning . . . . .	25
2.4.4 Mesh-width study . . . . .	27
2.4.5 Higher-order discretization of quasi free-surface boundary condition . . . . .	30
2.4.6 Case with increased amplitude . . . . .	32
2.5 Modified algorithm and numerical results . . . . .	32
2.5.1 The modified algorithm . . . . .	32
2.5.2 First results . . . . .	33
2.5.3 Further results . . . . .	36
2.6 Conclusions . . . . .	40

<b>3</b>	<b>Evaluation of Nonuniform Meshes for 3D Navier-Stokes Flow</b>	<b>41</b>
3.1	Introduction . . . . .	41
3.2	Computational method . . . . .	44
3.3	Numerical results . . . . .	44
3.3.1	Laminar boundary layer . . . . .	45
3.3.2	Variation of Reynolds number on piecewise-uniform mesh . . . . .	51
3.3.3	Flat plate with free-surface perturbation . . . . .	57
3.4	Conclusions . . . . .	62
<b>4</b>	<b>Numerical Simulation of 3D Steady Free-Surface Flows around a Ship Hull</b>	<b>63</b>
4.1	Introduction . . . . .	63
4.2	Computational method . . . . .	64
4.2.1	Discretization in curvilinear domain . . . . .	64
4.2.2	Governing equations in curvilinear coordinates and discretization . . . . .	65
4.2.3	Remeshing algorithm . . . . .	67
4.3	Numerical results . . . . .	69
4.3.1	Two approximations of free-surface flow . . . . .	72
4.3.2	Nonlinear free-surface method with remeshing . . . . .	83
4.4	Conclusions . . . . .	88
<b>5</b>	<b>Analysis of Continuous and Semi-Discrete Water-Wave Problem</b>	<b>91</b>
5.1	Introduction . . . . .	91
5.2	Analysis of continuous problem . . . . .	91
5.2.1	Fourier analysis of full Navier-Stokes equations . . . . .	92
5.2.2	Fourier analysis of reduced Navier-Stokes equations . . . . .	94
5.3	Analysis of semi-discrete problem . . . . .	96
5.3.1	For second-order accurate discretization of quasi free-surface condition . . . . .	96
5.3.2	For first-order upwind discretization of quasi free-surface condition . . . . .	99
5.4	Conclusions . . . . .	100
<b>6</b>	<b>Linearized Godunov Scheme for Two-Fluid Flow Computations</b>	<b>105</b>
6.1	Introduction . . . . .	105
6.2	Flow model . . . . .	106
6.2.1	Conservation equations . . . . .	106
6.2.2	Level-set equation . . . . .	107
6.2.3	Equations of state . . . . .	109
6.3	Discretization . . . . .	110
6.3.1	Finite volumes . . . . .	110
6.3.2	Riemann-problem approach . . . . .	112
6.3.3	Two-fluid, linearized Godunov scheme . . . . .	113
6.3.4	Boundary-condition treatment . . . . .	116
6.4	The two-fluid Godunov scheme for Tait's equation of state . . . . .	118

---

6.4.1	State space, Poisson curves and Hugoniot curves . . . . .	118
6.4.2	Families of Poisson curves . . . . .	119
6.4.3	Families of Hugoniot curves . . . . .	120
6.4.4	Intermediate states . . . . .	122
6.5	Conclusions . . . . .	124
<b>7</b>	<b>Fixes for Solution Errors near Two-Fluid Interfaces</b>	<b>125</b>
7.1	Introduction . . . . .	125
7.2	Error near interface . . . . .	126
7.2.1	Fluid-flow equations . . . . .	126
7.2.2	Analysis of flow model . . . . .	126
7.2.3	Guidelines for error improvements . . . . .	129
7.3	Fixes for error in cell with interface . . . . .	130
7.3.1	Advection of level-set function . . . . .	130
7.3.2	Advection of velocity, pressure and level-set function . . . . .	131
7.3.3	Advection of density and volume-of-fluid fraction . . . . .	131
7.3.4	Ghost-fluid method . . . . .	132
7.4	Numerical results . . . . .	134
7.4.1	Water front at constant speed and pressure . . . . .	134
7.4.2	Oscillating water column . . . . .	136
7.5	Conclusions . . . . .	144
<b>8</b>	<b>Computation of Gravity-Driven, Immiscible Water-Air Flows</b>	<b>145</b>
8.1	Introduction . . . . .	145
8.2	Equations of motion . . . . .	145
8.2.1	Fluid motion . . . . .	145
8.2.2	Interface motion . . . . .	146
8.3	Analysis . . . . .	147
8.3.1	Homogenization . . . . .	147
8.3.2	Steady-state solution property . . . . .	149
8.4	Numerical method . . . . .	149
8.5	Numerical results . . . . .	150
8.6	Conclusions . . . . .	156
<b>9</b>	<b>Conclusions and Recommendations</b>	<b>157</b>
9.1	Fitting . . . . .	157
9.2	Capturing . . . . .	159
	<b>Bibliography</b>	<b>161</b>
	<b>Samenvatting</b>	<b>169</b>
	<b>Curriculum Vitae</b>	<b>171</b>



# CHAPTER 1

## Introduction

### 1.1 Application area

The deployment of computational methods in the fluid-dynamics design cycle of modern transportation systems such as spacecraft, aircraft, cars and also ships, has shown a rapid increase, parallel with the progress in numerical mathematics and computer capability. At present, the use of computational fluid dynamics (CFD) in, e.g., ship design is such that, prior to any towing-tank model testing, the fluid-dynamics design of a ship's hull is analyzed by means of a computational method. If necessary, shortcomings of the design can thus be detected and corrected at an early stage. The large amount of data produced by a fluid-flow computation, allows us to gain a better insight in *(i)* topological flow properties such as flow separation, re-attachment, wave generation and vortex formation, as well as in *(ii)* quantitative data. Thus, the use of CFD methods results in better ship designs, within a shorter time span. In this way, CFD contributes to the economy of naval architecture and shipping, as well as to a reduction of pollution.

To assess the topological character of the flow around a ship one has to be aware of various complex flow features. For example, the flow around the ship's top structure is governed by aerodynamics and the underwater part by hydrodynamics, while both fluids are separated by a free surface. This free surface represents the wave pattern generated by the ship traveling at a certain speed (Figure 1.1). This free surface may show topological changes such as wave breaking and the formation of spray. In addition to this free surface other complex flow features are present, such as a stagnation region, a viscous region consisting of boundary layer and wake, a screw race, and in some cases lift effects on components such as rudder, keel and hydrofoils.

From a modeling point of view these features can be described by different mathematical models with varying degree of complexity. In the case of the free surface, the potential-flow model, with nonlinear boundary conditions, can be used to describe its (dynamic) behaviour. See, e.g., the classical texts [45, 50, 66, 80] for an elaborate exposition on the subject of free-surface flows. For the description of the flow in viscous regions, the boundary-layer or the Reynolds-averaged Navier-Stokes (RANS) equations are the proper mathematical models to consider. Within hydro-

dynamic research institutes, such as MARIN<sup>1</sup>, numerical methods for both features have matured well and are currently in routine use to assess hull designs. See, for example, [31, 60] for a description of the viscous flow solver PARNASSOS<sup>2</sup> and the free-surface potential-flow code RAPID<sup>3</sup>, respectively. PARNASSOS is a Navier-Stokes flow solver which does not compute free-surface gravity waves. RAPID is a free-surface-flow solver which does not compute viscous phenomena. This segregated approach is based on the validity of Froude's hypothesis, which states that the free surface and the viscous flow are only weakly coupled. However, to improve the degree of realism of the computational method the mutual influence of wave making and the viscous flow should be taken into account. This coupling of the free surface and the viscous bulk flows can be achieved by computing waves and viscous phenomena with a *single* flow solver: a Navier-Stokes solver which does allow for free-surface gravity waves. The present thesis is directed towards the efficient computation of two-fluid flows consisting of water and air, with as primary application area: ship hydrodynamics.



Figure 1.1: Example of the complex two-fluid flow to be addressed in numerical ship hydrodynamics (photo obtained from MARIN).

In an attempt to simulate the flow around a ship hull, it is imperative to realize that the above mentioned phenomena do not possess the same length and time scales. Depending on the length and time scales one is interested in, certain assumptions are made. In this thesis, the emphasis lies on the development of a

---

<sup>1</sup>[www.marin.nl](http://www.marin.nl)

<sup>2</sup>[www.marin.nl/original/services/softwaredevelopment/cph\\_parnassos.html](http://www.marin.nl/original/services/softwaredevelopment/cph_parnassos.html)

<sup>3</sup>[www.marin.nl/original/services/softwaredevelopment/cph\\_rapid.html](http://www.marin.nl/original/services/softwaredevelopment/cph_rapid.html)



single-fluid method which is capable of efficiently computing both the stationary wave pattern and the corresponding viscous water-flow regions generated by a ship traveling in still water. A computational method for two-fluid (water-air) flow is also proposed and tested in this thesis, though not yet for ship hulls. The inherent difficulty of free-surface flow problems is the interdependence of the free-surface location and the unknowns of the flow. In the present work we neglect the influence of the propeller, any effects of lifting surfaces, wave breaking or the formation of spray.

In Section 1.2, the two basic techniques used in this thesis for the representation of moving surfaces, i.e., the Eulerian and Lagrangean, are discussed. Section 1.3 gives an outline of this thesis, briefly introducing the topics of the various chapters.

## 1.2 Basic grid techniques

To compute a free surface on a computational grid, basically two techniques exist: (i) moving-grid or fitting (Lagrangean) techniques and (ii) fixed-grid (Eulerian) techniques. In the first, the free surface forms part of the boundary of the computational domain (the upper boundary in Figure 1.2a). In the second, the free surface is not aligned with the grid, see Figure 1.2b. Because in the moving-grid approach the free surface is a part of the boundary of the computational domain, its resolution on the grid is crisp, but it is not necessarily accurate; its location and topology may be poorly resolved. A moving-grid technique is not well-suited for the computation of bifurcating free surfaces. In practice, it will already be unsuited for free surfaces that are strongly distorted. Fixed-grid techniques may not suffer from such drawbacks, but here the front may be diffused. Mixed Lagrangean-Eulerian methods, which try to combine the best of the above two basic techniques, may be constructed. An example of such a hybrid approach would be to overlay a fixed grid with a narrow moving grid attached to the free surface (Figure 1.2c). Such complex combined grids, however, will not be considered in this thesis. We proceed by considering moving- and fixed-grid techniques in more detail.

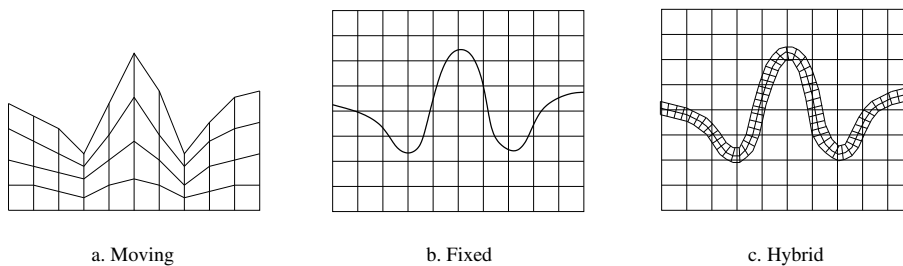


Figure 1.2: Types of grid techniques for computing free-surface flows.

### 1.2.1 Moving-grid techniques

A common approach in moving-grid techniques is to describe the free surface as a height function, say as  $h(\mathbf{x}, t) : \mathbb{R}^d \times \mathbb{R}^+ \rightarrow \mathbb{R}$ , with  $d = 1$  (Figure 1.3) or  $d = 2$ . Some references to free-surface height methods are [23, 53, 60]. It is common use to do the calculation of the free-surface height alternately with that of the bulk flow, the so-called *partitioned approach*, a reason being that in case of a single-valued height function, the free-surface-flow problem is one dimension lower than the bulk-flow problem. Methods that compute the height function and the bulk flow simultaneously do exist though, see, e.g., [6]. Such methods follow what is called the *monolithic approach*.

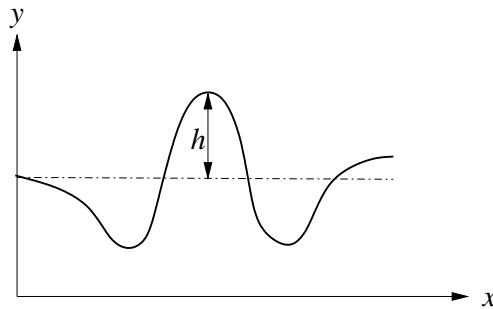


Figure 1.3: Free-surface height  $h$ .

The number of conditions to be imposed on the free-surface boundary is one more than on a fixed surface; it is equal to the usual number for the bulk flow ( $d$  conditions for the full incompressible Navier-Stokes equations in  $\mathbb{R}^d$ , e.g.) plus one extra for the height function. For free-surface-flow problems as occurring in ship hydrodynamics, the free-surface is a material surface and, hence, allows no mass transport through it, i.e.,  $\mathbf{u} \cdot \mathbf{n} = 0$ , where  $\mathbf{n}$  is the free-surface (unit) normal. It is furthermore assumed that the stresses at the free surface vanish. Imposing these conditions in a physically correct way is of paramount importance for a good resolution of the free surface and is not trivial. One subtlety in case of, e.g., steady free-surface water waves is that these are known to satisfy a dispersion relation (Chapter 3 in [50]), which uniquely relates the length of the free-surface wave to the Froude number. Ideally, in the discrete case, this dispersion relation should be satisfied as well. Another point of attention is the ‘contact-line’ problem, i.e., the intersection of the free surface and a no-slip boundary; an example of this is the water line at a ship’s hull. Here, the difficulty is that in the unsteady case the free-surface and the no-slip boundary conditions do not match.

Because the grid may strongly distort during the free surface’s motion, discretization of the flow equations in the computational domain is more appropriate than discretization in the physical domain. As a consequence it is harder to build in physics at a low discretization level because of the more intricate equations (due to the metric terms).

For steady free-surface-flow problems, a common ‘partitioned’ solution approach with the free-surface-height method is the following two-step process:

1. Solve the steady bulk-flow equations (the full Navier-Stokes equations, e.g.) with the free surface frozen and with – consequently – one free-surface condition not imposed.
2. On the basis of the previously obtained bulk-flow solution, correct the free surface such that it (better) satisfies the free-surface condition not imposed in the first step.  
If not all equations are sufficiently satisfied, return to step 1.

A pitfall in making this solution process efficient is to make both steps efficient separately, ignoring their interaction. This holds in particular for step 2: introducing large free-surface updates may hamper or even ruin the bulk-flow problem’s convergence to a steady state. For that reason, the rule mostly followed in steady free-surface flow computations is to carefully time-march the unsteady free-surface and bulk-flow equations to a steady state. However, because of the persistent unsteady free-surface-flow phenomena, this evolutionary approach is not efficient at all. In [59], still following the abovementioned two-step process, an efficient method is proposed for solving the steady free-surface flow (step 2) while keeping the induced perturbations in the bulk-flow computation (step 1) small. The key lies in the implementation of the free-surface conditions: in [59], a so-called quasi-free-surface condition is proposed. For smooth, single-valued free surfaces, this method appears to work perfectly well [14, 15, 59]. However, when steepening the water waves, as expected, its performance deteriorates and finally breaks down.

In case of unsteady free-surface flows, the technique of alternately updating the bulk flow and the free surface may be followed as well. However, in general this subcycling approach is only first-order accurate in time, independent of the accuracies of the separate time integrators for bulk flow and free-surface flow. (Evidence for this, in the application area of fluid-structure interactions, is given in [58].)

Fixed-grid methods hold out the promise of resolving a much larger class of free-surface flows and – also – of not suffering from a time-accuracy barrier in the unsteady case. In the next section, we consider some pros and cons of fixed-grid techniques.

### 1.2.2 Fixed-grid techniques

In the Eulerian approach, since many years, some well-proven techniques exist for computing flows with free boundaries. A classical method is the Marker-and-Cell (MAC) method [27]. Here, one of the two fluids is seeded with massless particles, that are passively advected with the flow. A grid cell without any particle is defined to be a cell fully filled with the other fluid (possibly void) and the free surface(s) may be defined as the set(s) of cell faces separating the cells with particles from the cells without, or – more accurately – as the tight contour(s) wrapped around the particles such that no cell without particles is closed in (Figure 1.4a). Flow bifurcations cause no difficulties for the MAC method. A deficiency though is that

no clear distinction can be made between physical and numerical cavitation (Figure 1.4a: empty cell at the bottom of the fluid). To avoid this uncertainty, the rule is to seed the fluid with as many particles as possible. Doing so, particularly in 3D, the MAC method may become very expensive.

A more efficient fixed-grid method is the Volume-of-Fluid (VOF) method [30]. In it, per grid cell, in addition to the standard fluid-flow quantities such as velocity and pressure, a scalar quantity is introduced, which represents the fraction of that cell, denoted by  $\alpha(\mathbf{x}, t) : \mathbb{R}^d \times \mathbb{R}^+ \rightarrow [0, 1]$ , that is filled with one of the two fluids: the VOF fraction (Figure 1.4b). The fraction is transported (in the incompressible case) by means of the advection equation for  $\alpha(\mathbf{x}, t)$ , i.e.,

$$\frac{\partial \alpha}{\partial t} + \mathbf{u} \cdot \nabla \alpha = 0. \quad (1.1)$$

The location of the free surface can be defined in a similar way as in the MAC method, viz. as the set of cell faces separating the cells with the VOF fraction larger than zero from the cells with the VOF fraction equal to zero, or – more accurately – at the subcell level. The latter requires intricate flux algorithms, particularly in 3D. A principal drawback of the VOF method is that the VOF fraction is non-smooth and – hence – hard to accurately resolve in precisely the region of interest: at the free surface.

A natural fix is to replace the VOF fraction by a smooth scalar function, which represents, e.g., the distance to the free surface. Doing so, the free surface is simply defined as the zero-level iso-surface. This is known as a level-set method (Figure 1.4c). Text books on level-set methods are [54, 65], a classical journal paper is [52]. Older work in which the technique is already found, though not yet under the name of level-set technique, is that by Markstein on flame propagation. See, e.g., [51], in which a flame front is represented by a higher-dimensional, differentiable function that is advected by the flow. Since last decade, level-set methods enjoy many publications, a list related to CFD only is [4, 8, 13, 18, 19, 20, 22, 28, 32, 41, 44, 57, 68, 69, 70, 83, 84]. When computing the flow of a single fluid with free surface, i.e., a flow with material-void interface, a minor problem of level-set methods is that no velocity field is defined in the void region. Hence, to guarantee smoothness of the level-set function, a proper artificial velocity field in the void region needs to be defined. This difficulty may also be seen as a good opportunity though to improve the free boundary's resolution without being inhibited by physics. E.g., in the void region an artificial velocity might be chosen which counteracts the effects of numerical diffusion by anti-advection [1, 44].

Keeping the level-set function smooth forms a point of attention. During the computation, the function may need to be regularized. In this reinitialization step, care needs to be taken that the free-surface location is preserved.

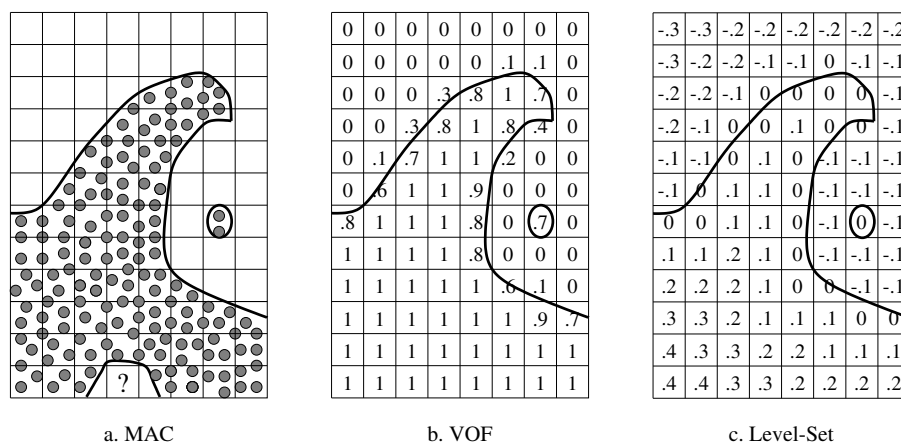


Figure 1.4: Known types of fixed-grid techniques for computing free-surface flows.

Using an MAC, a VOF or a level-set method, free-surface boundary conditions can be imposed at all cell faces separating the two fluids or – alternatively – at the subcell level. With a level-set method, the latter can be done more accurately than with the VOF method, but is still intricate, particularly in 3D. In [38], the approach of imposing free-surface conditions in the interior of the computational domain is called tracking. The free surface is genuinely free when it is *captured* (i.e., when no conditions at all are imposed on it). In capturing methods, in principle, the free surface is a two-fluid interface (with one fluid possibly virtual). The challenge of capturing methods is to choose or devise a physically correct two-fluid flux formula to be applied at the low discrete level of cell faces. A specific, well-known problem of capturing (not of tracking and fitting) techniques is that large solution errors may occur near the interface: the so-called ‘pressure-oscillation problem’, which has been observed first by Karni [35] and Abgrall [2]. In literature, several remedies against this have already been proposed.

As mentioned, both tracking and capturing can be combined with the MAC, the VOF as well as the level-set method. As compared to the surface-height method, with the MAC, VOF and level-set methods, computational overhead is introduced by the representation of the  $n$ -dimensional free surface as an  $(n + 1)$ -dimensional set of particles (MAC) or function (VOF and level-set). When computing a single-fluid flow problem with free surface by a capturing technique, even more overhead may be introduced by the computation of the flow of the virtual secondary fluid. Of course, the overhead should be and can be minimized. Possibly still remaining overhead is expected to be counterbalanced to a sufficient degree by the advantages of capturing: no remeshing, no limitation with respect to free-surface topologies, no free-surface boundary conditions to be imposed (the dispersion relations in case of water waves under the action of gravity, e.g., will be satisfied automatically), etc.

## 1.3 Outline

This section describes the main content of this thesis and can serve as a quick reference for the outline of each chapter. The thesis can be divided into two parts: *Part A*, consisting of Chapters 2 – 5 and containing the results of research on a *fitting* method, and *Part B*, consisting of Chapters 6 – 8 and containing the results of work done on a *capturing* method.

### 1.3.1 Part A: fitting method

Chapter 2 serves as an introduction to the free-surface fitting method which is investigated in this thesis. It contains a detailed description of the free-surface iteration method, based on the *quasi free-surface condition*. This iteration method requires the solution of a number of incompressible RANS boundary-value problems. The discretization and solution method of the RANS method are also described. The two topics of this chapter are (i) the extension to 3D of the existing 2D free-surface iteration method, described to full extent in [12], and (ii) the further improvement of the convergence of this iteration method. The convergence behaviour of the method is investigated by computing the free-surface and bulk flow for a problem in which a pressure perturbation is imposed on the free surface. Variation of the strength of the perturbation changes the nonlinearity of the resulting wave pattern. Various amplitudes of the perturbation are considered. This chapter has originally appeared as [47] and various separate results have already been published in [48, 49].

In Chapter 3 the perturbed uniform flow test case of Chapter 2 is extended by inserting a flat plate with unit length into the flow. The presence of the flat plate induces a boundary layer and a wake. The computation of this boundary layer and wake requires a non-uniform mesh. Two types of non-uniform meshes are considered here, namely the piecewise-uniform mesh, with as its known variant the *Shishkin mesh*, and the exponentially stretched mesh. The aim of this chapter is to establish the accuracy and convergence of our iterative method on the piecewise-uniform mesh in comparison with those on the more familiar exponentially stretched mesh. A mesh width variation and a Reynolds number variation are considered. The work done in this chapter has been presented by the author in an invited lecture at the Workshop on Numerical Methods for Singularly Perturbed Differential Equations, held at the Department of Mathematics and Statistics of the University of Limerick (Ireland) in 2002.

In Chapter 4 the free-surface iteration method, introduced in Chapter 2, is used to compute a ship hydrodynamics problem, i.e., the computation of the shape of the free surface, including the underlying bulk flow, generated by a Series 60 type hull traveling at a certain speed. A part of the results from this chapter has appeared in [49].

In Chapter 5, the fitting approach, investigated and further developed in Chapters 2 – 4, is analyzed with respect to (i) the existence of steady free-surface waves and (ii) the dispersion properties of these waves for two different discretizations of the quasi free-surface condition. The analysis is done for the *exact* Navier-Stokes equations (both full and reduced), as well as for the *modified* Navier-Stokes equa-

tions (only reduced). The modified equations correspond with the discretization method applied in Chapters 2 – 4. The analysis makes use of linearized equations, a Fourier technique, and the assumption of infinitely deep water and periodicity in mainstream direction. The analysis leads to a remarkable difference between the full and reduced Navier-Stokes equations, and yields a requirement for the existence of steady waves for the latter equations. Further, it explains the dispersion behaviour of the free-surface waves, as observed in our numerical results.

### 1.3.2 Part B: capturing method

Chapter 6 is based on [42]. In this chapter we derive a system of hyperbolic conservation laws for (still) compressible water-air flow, using a bulk-density formulation. As the equation of state for the two fluids, Tait's equation is used. Another equation to be specified describes the location of the interface. The level-set method is used for this purpose. Application of a finite-volume discretization leads to a Riemann problem at each finite-volume wall. For the (approximate) solution of this, we derive a two-fluid linearized Godunov scheme.

In Chapter 7 we continue the analysis of two-fluid flow problems and in particular the advection of a contact discontinuity separating two compressible and immiscible fluids. Also in this case the two fluids represent water and air. It is shown that already the trivial flow problem of a water-air interface running through a tube at constant speed and pressure poses considerable difficulties for most numerical schemes. Most numerical schemes suffer from the *pressure-oscillation problem*, i.e., an  $\mathcal{O}(h^0)$  accurate, non-monotone behaviour of the pressure across a contact discontinuity. An analysis is presented which uncovers the origin of this phenomenon. Several fixes for this zeroth-order accuracy problem are numerically investigated. Based on the numerical results we select the *ghost-fluid* method as object for further study. This chapter has appeared as [43] and a condensed version of Chapters 6 and 7 also appeared as [41].

In Chapter 8 the hyperbolic system, proposed in Chapter 6, is extended with a source term. This source term is added to represent the effect of gravity. An analysis of the non-homogeneous system of equations is presented. Next, the system of equations is homogenized. An analysis of the homogenized equations leads to an elegant numerical method (a splitting method) valid for small time steps. As far as we know, this method is new. Numerical experiments are performed computing the flow in a tube with two fluids, e.g., water and air, driven by the gravity force. The discretization method is the finite-volume method where the linearized Godunov scheme, derived in Chapter 6, is applied as approximate Riemann solver. The ghost-fluid method is used to resolve the solution in a cell containing the interface. These results have been published in [40].





## *Part A: Fitting Method*



## CHAPTER 2

# Efficient Computation of Steady, 3D Water-Wave Patterns

### 2.1 Introduction

In the case of a smooth free surface without overturning waves, i.e., when the free surface can be represented by a height function, surface-fitting methods are unsurpassed in accuracy. Because our primary focus is on the computation of non-overturning gravity waves, this method is adopted here.

The main topic of this chapter is the investigation of the convergence behavior for 3D problems, of a new non-monolithic (i.e., an alternating or partitioned) free-surface iteration method, proposed in [59] and worked out in [15] for 2D flows. In [11, 15, 59], the so-called quasi free-surface boundary condition is derived. In the present chapter, this boundary condition is extended to 3D and the free-surface iteration method is applied to a test case involving stationary gravity waves induced by a pressure perturbation imposed at the free surface of a 3D water flow with a depth  $d \gg \lambda$ , the so-called waves on deep water, where  $\lambda$  is the free-surface wavelength. Varying the amplitude of the perturbation alters the nonlinearity of the resulting wave system. Each step of the free-surface iteration method involves the solution of a stationary Navier-Stokes boundary-value problem. In an improved version of the free-surface algorithm, the Navier-Stokes equations are not fully solved per free-surface iteration, but only corrected through a single relaxation sweep. The numerical results are compared with a solution of the potential-flow method from [60].

The contents of this chapter are the following. In Section 2.2, the governing equations are introduced. Section 2.3 describes the computational method, in particular the stationary Navier-Stokes boundary-value problem which is solved in each free-surface iteration step. In Section 2.4, various numerical results are presented for the free-surface algorithm. In Section 2.5, the aforementioned improvement is made to the algorithm and corresponding numerical results are presented. Section 2.6 concludes this chapter.

## 2.2 Governing equations and free-surface conditions

In this section an outline is given of the mathematical model which describes viscous free-surface flows. The first subsection lists the equations which describe the fluid flow, the next subsection treats the free-surface conditions and the last the so-called quasi free-surface (boundary) condition.

### 2.2.1 Flow equations

Let  $\Omega \subset \mathbb{R}^3$  be the physical domain which is occupied by an incompressible, viscous fluid and let us split the domain boundary as  $\partial\Omega = \Gamma_{\text{FS}} \cup \Gamma_0$ , where  $\Gamma_{\text{FS}}$  denotes the free surface and  $\Gamma_0$  the remaining (fixed) part of  $\partial\Omega$ . Positions in  $\mathbb{R}^3$  are identified with respect to a Cartesian coordinate system  $\mathbf{e}_\alpha$ ,  $\alpha = \{x, y, z\}$  with  $\mathbf{g} = -g\mathbf{e}_z$  denoting the gravity force. The state of the flow is characterized by the velocity field  $\mathbf{u}(\mathbf{x}) : \Omega \rightarrow \mathbb{R}^3$  and the pressure  $p(\mathbf{x}) : \Omega \rightarrow \mathbb{R}$ . Incompressibility implies a solenoidal velocity field:

$$\nabla \cdot \mathbf{u} = 0, \quad \forall \mathbf{x} \in \Omega. \quad (2.1)$$

Conservation of momentum of the fluid is described by

$$\nabla \cdot (\mathbf{u}\mathbf{u}^T) + \nabla\varphi - \nabla \cdot (\varepsilon(\mathbf{x}) (\nabla\mathbf{u} + \nabla\mathbf{u}^T)) = \mathbf{0}, \quad \forall \mathbf{x} \in \Omega, \quad (2.2)$$

where  $\varepsilon(\mathbf{x}) = \text{Re}^{-1} + \nu_t(\mathbf{x})$  the viscosity parameter with  $\text{Re} = U\ell/\nu$  the Reynolds number and  $\nu_t(\mathbf{x})$  the turbulent eddy viscosity. Furthermore  $U$  and  $\ell$  are the reference speed and length and  $\nu$  is the kinematic viscosity. In this chapter we assume laminar flow, herewith setting  $\nu_t(\mathbf{x}) = 0$ . The unknown  $\varphi$  is the hydrodynamic pressure

$$\varphi(\mathbf{x}) := p(\mathbf{x}) + \text{Fr}^{-2}z, \quad (2.3)$$

which contains the second similarity parameter in viscous free-surface flows, the Froude number,  $\text{Fr} \equiv U/\sqrt{g\ell}$ , with  $g$  the acceleration of gravity. It is assumed that diffusion in main-flow direction, say the  $x$ -direction, can be safely neglected. As a result the viscous operator, on an orthogonal mesh and for a spatially invariant eddy-viscosity, reduces to  $\tilde{\Delta} := \partial_y^2 + \partial_z^2$ , which in turn reduces the number of boundary conditions to be imposed at the  $x$ -outlet boundary.

### 2.2.2 Free-surface conditions

The free-surface (boundary) conditions follow from the general interface conditions and the assumptions that both density and viscosity of one of the adjacent fluids vanish at the interface and that the interface is impermeable. In many applications of interest, especially in those which admit steady solutions, the free surface can be expressed as a single-valued height function  $\eta(x, y) : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ , denoted as  $\Gamma_{\text{FS}} = \{(\mathbf{x}) : (x, y, z = \eta(x, y))\}$ . Impermeability leads to the kinematic condition

$$\mathbf{u} \cdot \nabla\eta = \mathbf{u} \cdot \mathbf{e}_z. \quad (2.4)$$

This formulation imposes smoothness restrictions on the shape of the free surface. Vanishing interfacial stresses result in three dynamic conditions, namely

$$p(\mathbf{x}) - 2\text{Re}^{-1} \frac{\partial \mathbf{u}_n}{\partial \mathbf{n}} = p_{\text{FS}}(\mathbf{x}), \quad (2.5)$$

in the direction normal to the free surface, with  $p_{\text{FS}}(\mathbf{x})$  the specified pressure perturbation along the free surface, and

$$\mathbf{t}^\alpha \cdot \boldsymbol{\tau}(\mathbf{u}) \cdot \mathbf{n} = 0, \quad \alpha = 1, 2 \quad (2.6)$$

tangential to the free surface, where  $\boldsymbol{\tau}(\mathbf{u})$  is the viscous stress tensor for an incompressible fluid. Here  $\mathbf{n}$  and  $\mathbf{t}^\alpha$  ( $\alpha = 1, 2$ ) are the unit normal vector and the orthonormal tangential vectors, respectively. For the practical application envisaged here, the viscous contribution to the normal dynamic condition may be neglected, resulting in an inhomogeneous Dirichlet condition for the pressure  $p(\mathbf{x})$ . It has been assumed that surface tension effects can be safely ignored.

### 2.2.3 Quasi free-surface boundary condition

In free-surface flows, an interdependence exists of the state variables  $(\mathbf{u}(\mathbf{x}), p(\mathbf{x}))$  and their spatial domain, through both the kinematic and dynamic conditions. In general, the free-surface flow problem is stated by equations (2.1), (2.2) subject to (2.4)–(2.6) on  $\Gamma_{\text{FS}}$ , together with additional boundary conditions on  $\Gamma_0$ . Many concurrent free-surface iteration methods apply a formulation in which they first solve (2.1), (2.2), in time-dependent form, subject to the dynamic conditions at an approximate location of the free surface. In the following step, they adjust the free surface using the kinematic condition in time-dependent form. Results obtained with these methods can be found in, e.g., [5, 17] for the fully time-dependent approach and in, e.g., [23, 76] for pseudo-time integration and quasi-steady methods.

Here we apply a different free-surface iteration method based on the use of the quasi free-surface boundary condition which, in steady form yields

$$\mathbf{u} \cdot \nabla \varphi - \text{Fr}^{-2} \mathbf{u} \cdot \mathbf{e}_z = \mathbf{u} \cdot \nabla p_{\text{FS}}. \quad (2.7)$$

The derivation of this boundary condition in unsteady form, for  $p_{\text{FS}} = 0$ , can be found in [15]. Here, we will only employ the quasi free-surface condition in its steady form (2.7). Use of the quasi free-surface condition has the advantage that it does not decouple the kinematic and dynamic free-surface conditions. In fact, it is the combination of these kinematic and dynamic conditions which yields the wave-like solutions. The widely used counterpart of (2.7) is the kinematic boundary condition (2.4) in its unsteady form:

$$\eta_t + \mathbf{u} \cdot \nabla \eta = \mathbf{u} \cdot \mathbf{e}_z. \quad (2.8)$$

Use of this condition forces the flow equations to be in unsteady form, which may be principally unwanted for flows which are known to be steady. Moreover, it has been shown that the usual time integration method to solve for steady state is

computationally inefficient. This is due to the fact that the attenuation of gravity waves behaves like  $\mathcal{O}((t^{1-d})/2)$  in  $\mathbb{R}^d$ , see [15]. This is the main reason for directly addressing the steady formulation of the continuous problem. In this manner we circumvent the unfavourable temporal behaviour.

## 2.3 Computational method

In Section 2.3.1 we outline our iterative solution method for gravity-subjected free-surface flows. It requires the solution of a sequence of steady-state Navier-Stokes subproblems. A description of the latter is given in Section 2.3.2.

### 2.3.1 Free-surface iteration method

Denoting (2.1) and (2.2) as  $\mathcal{NS}(\mathbf{u}(\mathbf{x}), \varphi(\mathbf{x})) = 0$ , and the boundary conditions to be imposed at  $\Gamma_0$  as  $\mathcal{B}(\mathbf{u}(\mathbf{x}), \varphi(\mathbf{x})) = 0$ , the solution of the free-surface flow problem can be found by iterating the following two steps:

- I. For a given boundary  $\Gamma_{\text{FS}}$ , solve  $(\mathbf{u}(\mathbf{x}), \varphi(\mathbf{x}))^T$  from

$$\begin{aligned} \mathcal{NS}(\mathbf{u}(\mathbf{x}), \varphi(\mathbf{x})) &= 0, \quad \forall \mathbf{x} \in \Omega, \\ \mathcal{B}(\mathbf{u}(\mathbf{x}), \varphi(\mathbf{x})) &= 0, \quad \forall \mathbf{x} \in \Gamma_0, \end{aligned}$$

$$\left. \begin{aligned} \mathbf{t}^{(\alpha)} \cdot \boldsymbol{\tau}(\mathbf{u}) \cdot \mathbf{n} &= 0, \quad \alpha = 1, 2 \\ \mathbf{u} \cdot \nabla \varphi - \text{Fr}^{-2} \mathbf{u} \cdot \mathbf{e}_z &= \mathbf{u} \cdot \nabla p_{\text{FS}} \end{aligned} \right\}, \quad \forall \mathbf{x} \in \Gamma_{\text{FS}}.$$

If  $\|p - p_{\text{FS}}\| > \epsilon_{\text{FS}}$ , where  $\epsilon_{\text{FS}}$  is a suitably chosen small tolerance, then do step II, else stop.

- II. Use the solution  $(\mathbf{u}(\mathbf{x}), \varphi(\mathbf{x}))^T$  of I to obtain a new approximation of  $\Gamma_{\text{FS}}$  according to

$$\{(x, y, z := \text{Fr}^2(\varphi(\mathbf{x}) - p_{\text{FS}}(\mathbf{x})) : \forall \mathbf{x} \in \Gamma_{\text{FS}}\}, \quad (2.11)$$

next return to step I.

Note that at convergence  $p = p_{\text{FS}}$ .

### 2.3.2 Discrete Navier-Stokes boundary-value problem

The free-surface iteration method of the previous section requires the solution of the reduced Navier-Stokes boundary-value problem mentioned in step I. The discretization of this boundary-value problem is outlined first.

#### Discrete flow equations

Let  $\Omega_h$  be the orthogonal and equidistant partitioning of  $\bar{\Omega}$ , with  $\mathbf{x} = (i\Delta x, j\Delta y, k\Delta z)$ . On  $\Omega_h$  we denote a grid function  $f(i\Delta x, j\Delta y, k\Delta z)$  by  $f_{i,j,k}$ . At all grid points, both in the interior and at the boundaries, the derivatives are replaced by finite

differences as specified below. As mentioned,  $x$  is associated with the main-flow direction. In the continuity equation the first-order velocity derivative in the main-flow direction is discretized using an  $\mathcal{O}(h^2)$  upwind-difference scheme with fixed bias contrary to the first-order velocity derivative in the momentum equation which is upwind biased. The remaining first-order velocity derivatives in the continuity equation and the momentum equations are discretized using an  $\mathcal{O}(h^3)$  upwind-biased difference scheme with a fixed bias for the continuity equation. The pressure gradient is discretized using an  $\mathcal{O}(h^3)$  downwind-difference scheme with a fixed bias. The opposite biasing of the velocity divergence and the pressure gradient ensures discrete ellipticity, herewith avoiding ‘odd-even decoupling’ of the solution [31]. The second-order velocity derivatives are discretized with standard, second-order accurate central differences. In formula, the discrete continuity equation reads

$$D_x^- u_{i,j,k} + D_y^- v_{i,j,k} + D_z^- w_{i,j,k} = 0, \quad (2.12)$$

and the discrete momentum equations, for positive  $u_{i,j,k}$ ,  $v_{i,j,k}$  and  $w_{i,j,k}$ :

$$\begin{aligned} & u_{i,j,k} D_x^- u_{i,j,k} + v_{i,j,k} D_y^- u_{i,j,k} + w_{i,j,k} D_z^- u_{i,j,k} + \\ & D_x^+ \varphi_{i,j,k} - \text{Re}^{-1} (D_y^2 u_{i,j,k} + D_z^2 u_{i,j,k}) = 0, \end{aligned} \quad (2.13a)$$

$$\begin{aligned} & u_{i,j,k} D_x^- v_{i,j,k} + v_{i,j,k} D_y^- v_{i,j,k} + w_{i,j,k} D_z^- v_{i,j,k} + \\ & D_y^+ \varphi_{i,j,k} - \text{Re}^{-1} (D_x^2 v_{i,j,k} + D_z^2 v_{i,j,k}) = 0, \end{aligned} \quad (2.13b)$$

$$\begin{aligned} & u_{i,j,k} D_x^- w_{i,j,k} + v_{i,j,k} D_y^- w_{i,j,k} + w_{i,j,k} D_z^- w_{i,j,k} + \\ & D_z^+ \varphi_{i,j,k} - \text{Re}^{-1} (D_x^2 w_{i,j,k} + D_y^2 w_{i,j,k}) = 0, \end{aligned} \quad (2.13c)$$

where  $D_\alpha^\beta$  represents the following discrete differential operators

$$D_x^- = \frac{1}{\Delta x} \left( \frac{3}{2} E_x^0 - 2E_x^{-1} + \frac{1}{2} E_x^{-2} \right), \quad (2.14a)$$

$$D_\alpha^- = \frac{1}{\Delta \alpha} \left( \frac{1}{3} E_\alpha^{+1} + \frac{1}{2} E_\alpha^0 - E_\alpha^{-1} + \frac{1}{6} E_\alpha^{-2} \right), \quad \alpha = \{y, z\}, \quad (2.14b)$$

$$D_\alpha^+ = \frac{1}{\Delta \alpha} \left( -\frac{1}{3} E_\alpha^{-1} - \frac{1}{2} E_\alpha^0 + E_\alpha^{+1} - \frac{1}{6} E_\alpha^{+2} \right), \quad \alpha = \{x, y, z\}, \quad (2.14c)$$

$$D_\alpha^2 = \frac{1}{\Delta \alpha} (E_\alpha^{+1} - 2E_\alpha^0 + E_\alpha^{-1}), \quad \alpha = \{y, z\}, \quad (2.14d)$$

with the linear shift operator  $E_x^n$  defined as  $E_x^n u_{i,j,k} := u_{i+n,j,k}$ .

### Computational domain and boundary conditions

In  $\mathbb{R}^d$ , a full incompressible Navier-Stokes problem requires  $d$  boundary conditions on all boundary planes. Due to the reduced viscous operator, only one boundary condition may be imposed at an outlet boundary perpendicular to the  $x$ -direction. All other boundary planes require  $d$  boundary conditions, hereafter referred to as *physical* boundary conditions.

**Physical boundary conditions** The computational domain is defined by  $\bar{\Omega} = \{(x, y, z) : [x_{\text{in}}, x_{\text{out}}] \times [0, y_e] \times [z_b, \eta]\}$ . At the boundaries of  $\bar{\Omega}$  we impose the following conditions:

- **Inflow boundary** ( $x = x_{\text{in}}$ ). At this boundary three Dirichlet conditions are imposed in the form of specification of the velocity components, i.e.

$$\mathbf{u} = (U_\infty, 0, 0)^T. \quad (2.15)$$

- **Outflow boundary** ( $x = x_{\text{out}}$ ). Here an inhomogeneous Neumann condition is imposed on the hydrodynamic pressure, viz.

$$\varphi_x + uu_x = 0. \quad (2.16)$$

- **External boundary** ( $y = y_e$ ). Dirichlet conditions are imposed on the tangential velocity components  $u$  and  $w$ , and on the hydrodynamic pressure  $\varphi$ :

$$u = U_\infty, \quad w = 0, \quad \varphi = 0. \quad (2.17)$$

- **Internal boundary plane** ( $y = 0$ ). The internal boundary is taken as a symmetry boundary. Homogeneous Neumann conditions are imposed on the tangential velocity components and a likewise homogeneous Dirichlet condition on the normal velocity component:

$$\frac{\partial u}{\partial y} = 0, \quad v = 0, \quad \frac{\partial w}{\partial y} = 0. \quad (2.18)$$

- **Bottom** ( $z = z_b$ ). Also here, homogeneous Neumann conditions are imposed on the tangential velocity components, in this case  $u$  and  $v$ , and a homogeneous Dirichlet condition on the normal velocity component:

$$\frac{\partial u}{\partial z} = 0, \quad \frac{\partial v}{\partial z} = 0, \quad w = 0. \quad (2.19)$$

- **Free-surface boundary** ( $z = \eta$ ). At the free surface the following set of boundary conditions is imposed. Vanishing of the tangential stress components at the free surface results in

$$\frac{\partial u}{\partial n} = 0, \quad \frac{\partial v}{\partial n} = 0, \quad (2.20)$$

with  $n$  the normal to the free surface. Further, a Gaussian pressure perturbation with the following form is prescribed:

$$p_{\text{FS}}(x, y) = P e^{\alpha((x-x_c)^2 + (y-y_c)^2)}, \quad P > 0, \quad \alpha < 0. \quad (2.21)$$

The location  $(x_c, y_c)$  of the maximum pressure, as well as the parameters  $P$  and  $\alpha$  may be varied. Substituting (2.21) into the quasi free-surface boundary condition (2.7) leads to

$$\mathbf{u} \cdot \nabla \varphi - \text{Fr}^{-2} \mathbf{u} \cdot \mathbf{e}_z = 2\alpha p_{\text{FS}}(\mathbf{x})(\mathbf{u} \cdot \mathbf{r}), \quad \mathbf{r} = (x - x_c, y - y_c, 0)^T. \quad (2.22)$$

Note the nonlinearity of the quasi free-surface boundary condition.



All Neumann boundary conditions are discretized using  $\mathcal{O}(h^2)$  one-sided differences. The pressure gradient in the quasi free-surface boundary condition is discretized using an  $\mathcal{O}(h^2)$  upwind-difference scheme.

**Numerical boundary conditions** Because the discretization stencils extend beyond the boundaries of the computational domain, for the numerical solution procedure, numerical boundary conditions are required (for filling the virtual grid points outside the computational domain). For this purpose, quadratic solution extrapolation from the interior is applied.

### Solution method

The discretized equations form a system of nonlinear algebraic equations, which is solved by the SOR-Newton method. The sparsity pattern of the resulting linear algebraic system favors an iterative solution method over a direct method. The linear system is solved by adopting a space-marching Gauss-Seidel algorithm in which the marching is performed in the main-flow direction (the  $x$ -direction). This approach is natural in view of the strong parabolic nature of the flow equations. A particular feature of this solution method is that the system of equations is solved in a coupled manner. This approach avoids the use of an artificial-compressibility method or a SIMPLE-type algorithm.

The space-marching procedure yields a size reduction of the linear-algebra problem. The smaller linear systems are solved using a CILU(0) preconditioned Krylov-subspace method (GMRES). The search space for the Krylov method is limited to 20 vectors. The convergence criterion for the solution of the inner (Newton) iteration process is that the residuals of the bulk flow have all dropped below some very small threshold value. More details on the solution strategy of the system of algebraic equations are given in [31, 77]. The method has been implemented in the computer code PARNASSOS

## 2.4 Numerical results

In this section we present some numerical results obtained with the free-surface iteration method described in the previous section.

### 2.4.1 First results

As a first test case we consider the following parameter values for the Gaussian pressure perturbation (2.21):  $P = 0.05$ ,  $\alpha = -4$ ,  $\text{Fr} = 0.6$  and  $(x_c, y_c) = (0, 0)$ . This is conform to computations done with the potential-flow method described in [81].  $\text{Re}$  is set at  $10^6$ . The current computation is performed on the basis mesh,  $\Omega_h$ , which has 81, 31 and 31 nodes in the  $x$ -,  $y$ - and  $z$ -direction, respectively.  $\Omega_h$  is constructed such that the Kelvin wedge, which bounds the spatial distribution of the wave energy, does not intersect the external boundary by taking  $x_{\text{out}} = 6$  and  $y_e = 3$ . The wedge makes a semi-angle of  $19.5^\circ$  with the main-flow direction. (See Section 3.10 in [50] for a derivation of this result.) Other boundary coordinates

chosen are  $x_{\text{in}} = -2$  and  $z_b = -3$ . For all computations, the initial estimate of the free surface is the plane  $z = 0$ .

As mentioned before, our focus is on the convergence behavior of the new free-surface iteration method. We distinguish two iteration processes: an outer- and an inner-iteration process, the iteration processes II and I, respectively, as described in Section 2.3.1. The convergence of the outer iteration, the free-surface method, is measured through the pressure defect at the free surface. The convergence of the inner iteration, the Navier-Stokes method, is measured by computing  $\|\mathbf{R}\|_{\infty}$ , where  $\mathbf{R}$  is the residual vector of the flow equations (2.12),(2.13). For the present test case, the convergence behavior of the inner iteration is shown in the left graph of Figure 2.1.

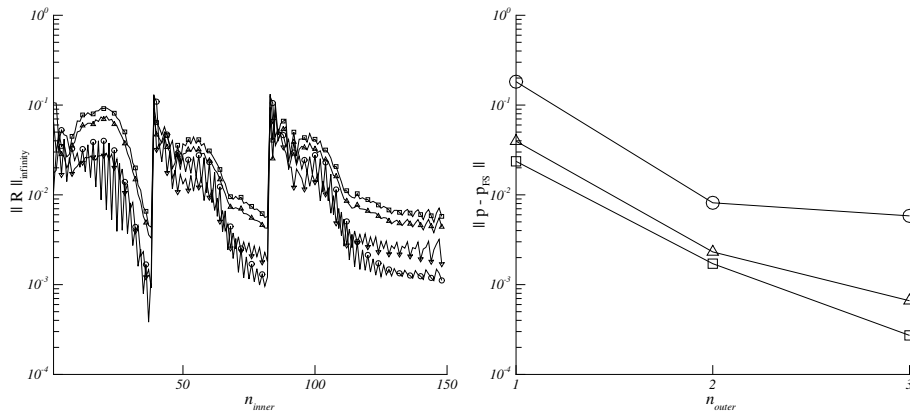


Figure 2.1: Convergence histories for the Gaussian pressure perturbation with  $P = 0.05$ , on  $\Omega_h$ . *Left*: of the inner-iteration process,  $\mathbf{R}$  is the residual of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ), the  $y$ -momentum equation ( $\circ$ ), and the  $z$ -momentum equation ( $\nabla$ ), only every fourth marker is shown. *Right*: of the free-surface pressure defect; measured in  $L_1$ -norm ( $\square$ ),  $L_2$ -norm ( $\Delta$ ), and  $L_{\infty}$ -norm ( $\circ$ ).

The two large jumps in the residual (at about  $n = 40$  and  $n = 80$ ) are due to free-surface updates. Figure 2.1 reveals that the inner-iteration process on the mesh obtained after the third free-surface update starts to oscillate with an increasing amplitude, preventing further decrease of the residuals. A closer inspection has shown that these residuals occur at a location near the outflow boundary, in the first grid plane underneath the free surface. This indicates a local incompatibility between the free-surface flow and the underlying bulk-flow solution. The proposed remedy will be addressed in the next section. The convergence of the free-surface iteration process is monitored through the pressure defect  $\|p^n - p_{\text{FS}}\|$ , measured in some usual norms. Here  $p^n = \varphi^n - \text{Fr}^{-2}z^n$  is the hydrodynamic pressure minus the hydrostatic part. The decrease in the pressure defect is shown in the right graph of Figure 2.1. The free-surface iteration appears to converge very fast; the second and third free-surface updates are already negligibly small as compared to the first, as can be seen in the left graph of Figure 2.2.

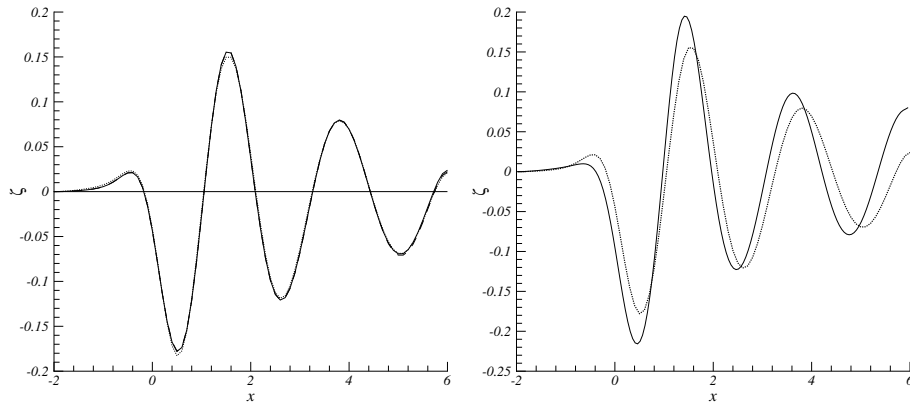


Figure 2.2: Wave elevations in the plane  $y = 0$  for the Gaussian pressure perturbation with  $P = 0.05$ . *Left:* of the present free-surface Navier-Stokes method on  $\Omega_h$ ; initial ( $\zeta = 0$ ),  $n = 1$  (dotted),  $n = 2$  (dashed), and  $n = 3$  (solid). *Right:* of the potential-flow method from [60] (solid) and the present Navier-Stokes method (dotted).

In Figure 2.2,  $\zeta = \eta/\eta_{max}$  is the wave elevation divided by the maximum obtainable elevation  $\eta_{max} = \frac{Fr^2}{2}$ . The last iterate in the left graph of Figure 2.2 shows a wavelength of  $\lambda = 2.3$  and a maximum scaled amplitude of about 15%. These results correspond fairly well with the results obtained through the potential-flow method described in [81]. For further comparison purposes a solution for this test case has also been computed through the potential-flow method described in [60]. The corresponding wave pattern is shown in the right graph of Figure 2.2, together with the present Navier-Stokes solution (the dotted line). Differences between both wave patterns are to be attributed to differences in the two continuous models and, particularly, to differences in their numerical discretizations. In Figure 2.3 we still show the entire Navier-Stokes wave pattern as obtained after the third free-surface update.

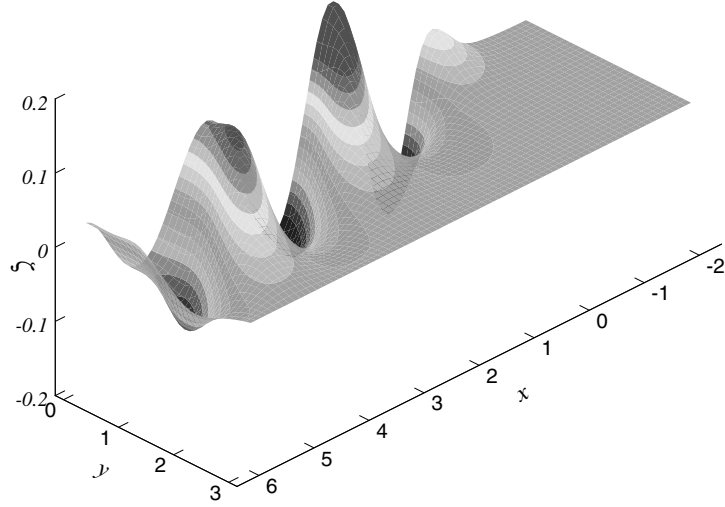


Figure 2.3: Wave elevation for the Gaussian pressure perturbation with  $P = 0.05$ , on  $\Omega_h$ , after three free-surface updates.

#### 2.4.2 Influence of outflow boundary condition

The starting divergence of the inner iteration on the third mesh, as shown in Figure 2.1, is due to an incompatibility between the free-surface flow solution and the underlying bulk-flow boundary condition at the outflow boundary. This discrepancy can be overcome by extending  $\Omega_h$  with a far-field wave-dissipation zone and a transition zone in between (see, e.g., [33, 36]). The wave-dissipation zone is added to rapidly dissipate all the wave energy from the numerical solution, hereby reducing the solution to uniform flow conditions.

The wave energy is better dissipated by increasing the numerical viscosity. This is achieved by (i) reducing the accuracy of the discretization of (2.7) in the wave-dissipation zone to first order, and (ii) by applying grid stretching in the wave-dissipation zone. The grid-point distribution in the wave-dissipation zone is controlled by

$$x_i = x_{\text{out}} e^{\alpha \frac{\Delta x}{x_{\text{out}}}} i, \quad y_j = y_e e^{\alpha \frac{\Delta y}{y_e}} j, \quad i = 1, \dots, (N_x)_{\text{add}}, \quad j = 1, \dots, (N_y)_{\text{add}}, \quad (2.23)$$

where  $\alpha$  is the stretching factor in  $x$ - and  $y$ -direction, and where  $(N_x)_{\text{add}}$  and  $(N_y)_{\text{add}}$  are the numbers of additional points in both directions. Both numbers are fixed by specifying the maximally allowable mesh width, e.g.,  $|x_{(N_x)_{\text{add}}} - x_{(N_x)_{\text{add}}-1}|$  is set at 0.5. The same is done for the  $y$ -direction. The extended mesh has  $149 \times 53 \times 31$  points, and is denoted by  $\hat{\Omega}_h$ . In absence of a wave solution near the outflow plane the incompatibility between the free-surface flow and the bulk flow disappears and, instead of (2.16), a homogeneous Neumann condition for the pressure can be specified as outflow boundary condition.

On this extended domain  $\hat{\Omega}_h$  we compute five free-surface updates. The positive influence of the wave-dissipation zone on the convergence behavior of both the inner and outer iteration can be seen when comparing Figure 2.4 with Figure 2.1.

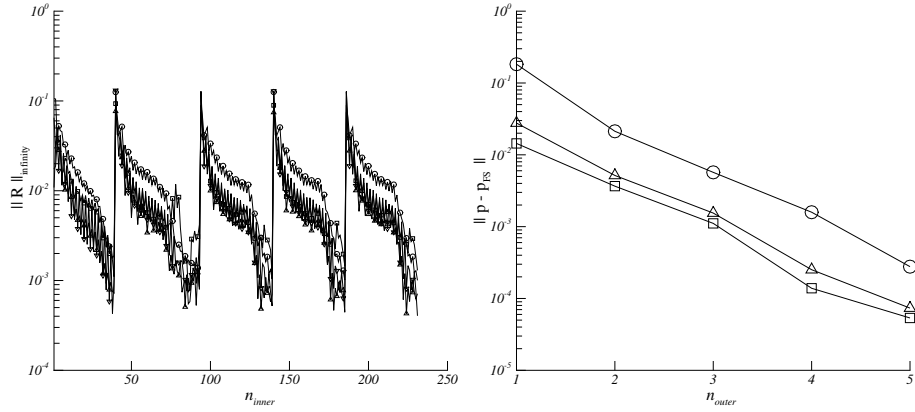


Figure 2.4: Convergence histories for the Gaussian pressure perturbation with  $P = 0.05$ , on  $\hat{\Omega}_h$ . *Left*: of the inner-iteration process,  $\mathbf{R}$  is the residual of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ), the  $y$ -momentum equation ( $\circ$ ), and the  $z$ -momentum equation ( $\nabla$ ), only every fourth marker is shown. *Right*: of the free-surface pressure defect; measured in  $L_1$ -norm ( $\square$ ),  $L_2$ -norm ( $\Delta$ ), and  $L_\infty$ -norm ( $\circ$ ).

Note again the jumps of  $\|\mathbf{R}\|_\infty$  after each free-surface update. These jumps are due to the fact that after each free-surface update the preconditioner, used in solving the stationary Navier-Stokes boundary-value problem, is reinitialized, i.e., its coefficients are set to zero. The effect of not reinitializing the preconditioner is studied in the next section.

In Figure 2.5, we show the entire wave pattern as obtained after the fifth free-surface update. The wave damping in the dissipation zone, which starts at  $x = 10$ , is clearly visible.

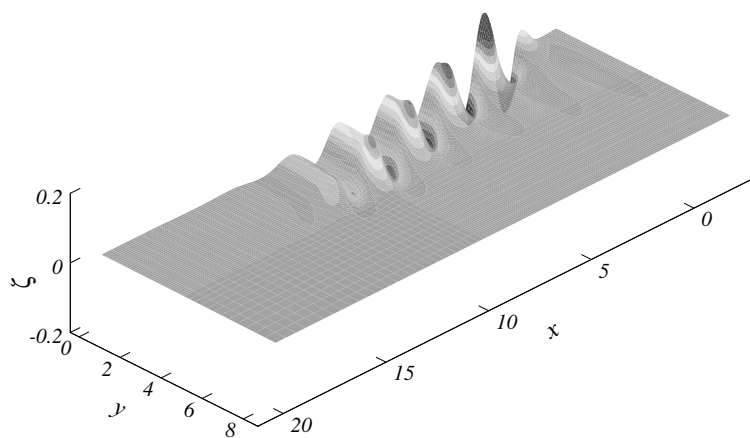


Figure 2.5: Wave elevation for the Gaussian pressure perturbation with  $P = 0.05$ , on  $\hat{\Omega}_h$ , after five free-surface updates.

In Figure 2.6, we still depict the elevations after the first and fifth free-surface update, in the planes  $y = 0$  and  $y = 2$ . Particularly from the left graph in Figure 2.6, it appears that the free-surface iteration converges very fast. (The initial estimate for the free surface is  $\zeta = 0$ .) Note that, as opposed to the first iterate in the left graph of Figure 2.6 and as opposed to the three iterates in the left graph of Figure 2.2, the free surface in the symmetry plane, as obtained after the fifth update, shows a small trough at about  $x = -1$ . The trough can still be observed at  $y = 2$  (the right graph of Figure 2.6).

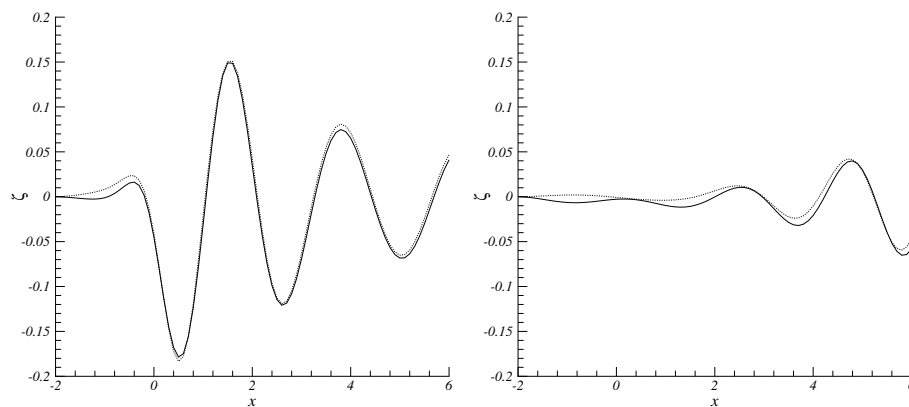


Figure 2.6: Wave elevation for the Gaussian pressure perturbation with  $P = 0.05$ , on  $\hat{\Omega}_h$ ;  $n = 1$  (dotted) and  $n = 5$  (solid). *Left:* In the plane  $y = 0$ . *Right:* In the plane  $y = 2$ .

In Figure 2.7 we show our Navier-Stokes wave patterns in the symmetry plane, as

obtained on the domains *with* and *without* wave-dissipation zone. For comparison purposes, in Figure 2.7 we also give the wave pattern obtained with the potential-flow method described in [60]. Differences between the two Navier-Stokes wave patterns in Figure 2.7 may be attributed mainly to the fact that the solution on the domain without wave-dissipation zone is less far converged than that on the domain with wave-dissipation zone. Still concerning Figure 2.7, note the strong wave damping starting from  $x = 10$ , the  $x$ -coordinate of the upstream boundary of the wave-dissipation zone.

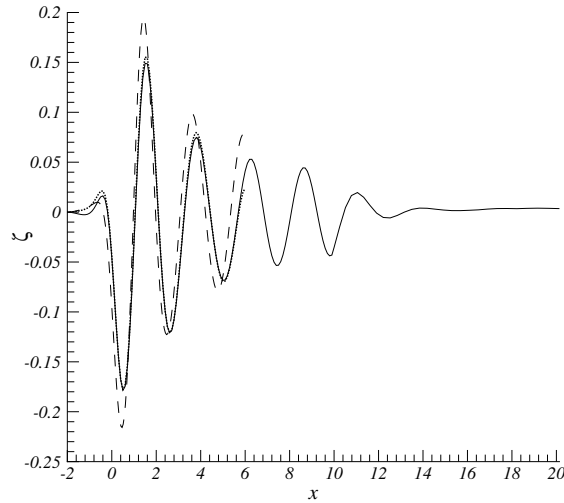


Figure 2.7: Wave elevation in the plane  $y = 0$  for the Gaussian pressure perturbation with  $P = 0.05$ ; present Navier-Stokes method, on  $\Omega_h$  and  $\hat{\Omega}_h$  (solid and dotted, respectively), and potential-flow method from [60] (dashed).

### 2.4.3 Influence of preconditioning

The convergence of  $\|\mathbf{R}\|_\infty$  reveals a jump after each free-surface update. It has been stated in the previous section that this is due to the reinitialization of the preconditioner, used to solve the stationary Navier-Stokes boundary-value problem. This section investigates the effect on the overall convergence behaviour of *not* reinitializing the preconditioner. The convergence behaviour, *without* reinitialization, is shown in Figure 2.8. Comparing Figures 2.4 and 2.8 reveals the benefit. In the remainder of this thesis the preconditioner is no longer reinitialized. The convergence rate of the pressure defect decreases after the second free-surface update, i.e., starting from  $n_{outer} = 3$ . It has to be noted that the norm of the free-surface pressure defect is taken over the whole free surface, i.e., including the wave-dissipation zone. An investigation of the spatial distribution of the pressure defect shows that this is concentrated in the latter area. Since we are only interested in the solution on  $\Omega_h$ , we may take the norm over this area only. Doing so, the convergence of the outer iteration shows a rapid decrease, see Figure 2.9. From Figure 2.9 it may be

concluded that the free-surface flow problem is solved after the second free-surface update, to within the accuracy  $\epsilon_{NS}$  specified in the software. Unless indicated otherwise, in the remainder of this chapter, the convergence of the inner and outer iteration is always measured over the entire computational domain, i.e., including the wave-dissipation zone.

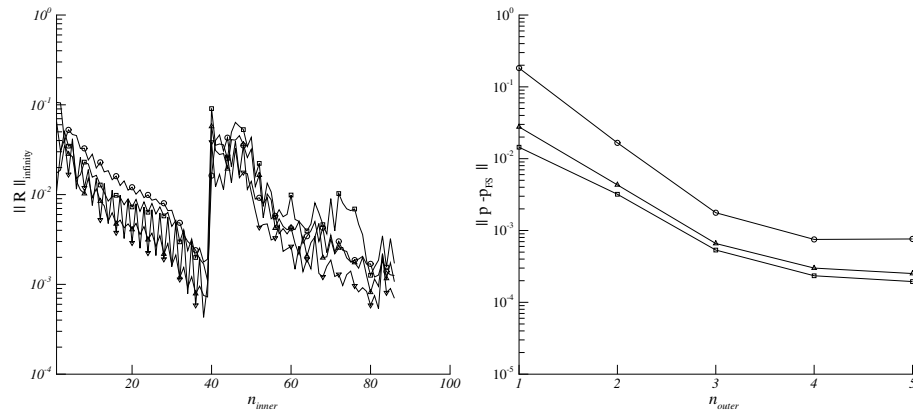


Figure 2.8: Convergence histories for the Gaussian pressure perturbation with  $P = 0.05$ , on  $\hat{\Omega}_h$ . *Left:* of the inner-iteration process,  $\mathbf{R}$  is the residual of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ), the  $y$ -momentum equation ( $\circ$ ), and the  $z$ -momentum equation ( $\nabla$ ), only every fourth marker is shown. *Right:* of the free-surface pressure defect; measured in  $L_1$ -norm ( $\square$ ),  $L_2$ -norm ( $\Delta$ ), and  $L_\infty$ -norm ( $\circ$ ).



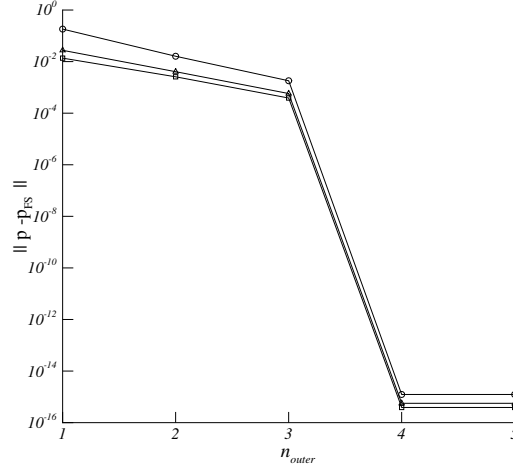


Figure 2.9: Convergence history of  $\|p - p_{FS}\|$  for the Gaussian pressure perturbation with  $P = 0.05$ , Flow computation on  $\hat{\Omega}_h$ , but  $\|p - p_{FS}\|$  measured on  $\Omega_h$ , in  $L_1$ -norm ( $\square$ ),  $L_2$ -norm ( $\triangle$ ) and  $L_\infty$ -norm ( $\circ$ ).

#### 2.4.4 Mesh-width study

In this section, the effects of the mesh width on the convergence and accuracy of the solution are investigated. This is done by comparing the numerical results obtained for  $\hat{\Omega}_h$  with results obtained for  $\hat{\Omega}_{2h}$  and  $\hat{\Omega}_{\frac{h}{2}}$ . (To reduce the influence of the outflow-boundary condition a wave-dissipation zone is added to each of the three grids.)

**Convergence behavior of inner and outer iteration** The convergence of the inner iteration on  $\hat{\Omega}_{2h}$  and  $\hat{\Omega}_{\frac{h}{2}}$  is shown in Figure 2.10. It appears that the strategy for the solution of the Navier-Stokes subproblems is not (yet) optimally efficient. To further illustrate this, in Figure 2.11, we have plotted  $n_{inner}$ , the number of inner iterations needed for the first Navier-Stokes subproblem, versus  $N_x$ , the number of grid points in  $x$ -direction. A least-squares fit reveals a linear dependence of  $n_{inner}$  on  $N_x$ . (Ideally,  $n_{inner}$  is independent of  $N_x$ ; this might be realized with a proper multigrid method.)

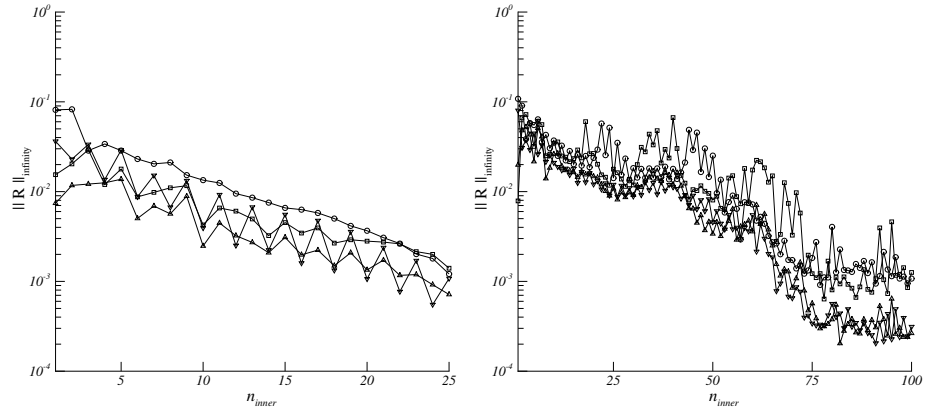


Figure 2.10: Convergence histories of the inner-iteration process for the Gaussian pressure perturbation with  $P = 0.05$ ,  $\mathbf{R}$  is the residual of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ), the  $y$ -momentum equation ( $\circ$ ), and the  $z$ -momentum equation ( $\nabla$ ), only every fourth marker is shown. *Left:* on  $\hat{\Omega}_{2h}$ . *Right:* on  $\hat{\Omega}_{\frac{h}{2}}$ .

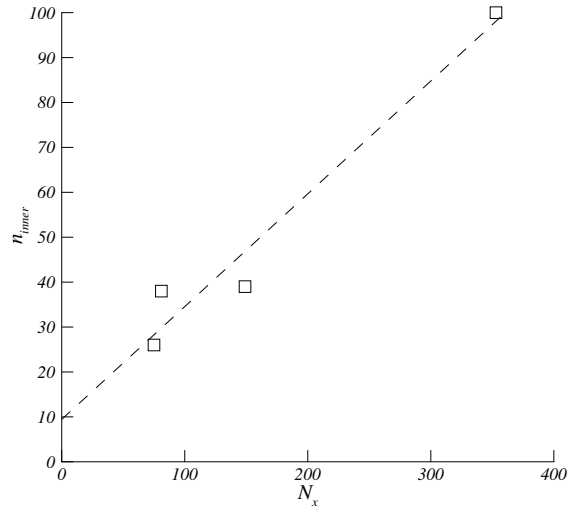


Figure 2.11: Required number of inner iterations in first Navier-Stokes subproblem versus the number of grid points in  $x$ -direction.

The outer iteration's convergence behavior on the three grids ( $\hat{\Omega}_{2h}$ ,  $\hat{\Omega}_h$ ,  $\hat{\Omega}_{\frac{h}{2}}$ ) is shown in Figure 2.12. Comparing the three convergence rates from  $n_{outer} = 2$  to  $n_{outer} = 3$ , it seems that the convergence is fairly grid-independent.

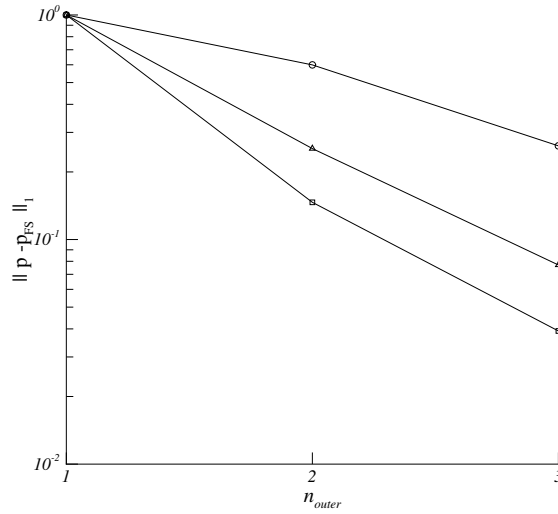


Figure 2.12: Convergence histories of the outer-iteration process for the Gaussian pressure perturbation with  $P = 0.05$ , measured in  $L_1$ -norm;  $\hat{\Omega}_{2h}$  ( $\square$ ),  $\hat{\Omega}_h$  ( $\Delta$ ), and  $\hat{\Omega}_{\frac{h}{2}}$  ( $\circ$ ).

**Accuracy behavior** For the three grids, the wave elevations in the plane  $y = 0$  are shown in Figure 2.13. They show a clear dependence of the wavelength and wave amplitude on the mesh width. These effects can be attributed to the discretization of the quasi free-surface boundary condition and the flow equations. In (2.7),  $\mathbf{u} \cdot \nabla \varphi$  is discretized using the  $\mathcal{O}(h^2)$  upwind scheme. First note that (2.22) is equivalent with

$$\mathbf{u} \cdot \nabla p = \mathbf{u} \cdot \nabla p_{\text{FS}}, \quad (2.24)$$

using the definition (2.3). The mesh-width dependence can be partially understood through a spectral analysis of (2.7). The term in (2.7) most relevant for the wavelength in the main-flow direction is  $up_x$ . For first analysis purposes, linearizing around the unperturbed flow, (2.7) can be reduced to  $up_x = 0$ . For  $u$  positive, the corresponding modified equation reads

$$u \frac{\partial p}{\partial x} = u \frac{h^2}{3} \frac{\partial^3 p}{\partial x^3} - u \frac{h^3}{4} \frac{\partial^4 p}{\partial x^4} + \mathcal{O}(h^4). \quad (2.25)$$

Inserting a single wave solution of the form  $p = P e^{ikx}$  into (2.25), results in

$$uik \left( 1 + \frac{1}{3} h^2 k^2 \right) P e^{ikx} = \mathcal{O}(h^3). \quad (2.26)$$

From this relation it can be concluded that the numerical advection velocity of  $p$  increases with increasing mesh width, thus increasing the length of the gravity wave. The latter increase can be explained from the dispersion relation for waves on deep water, see Chapter 3 in [50]. Particularly when applying a multigrid solution strategy, one should be aware of the mesh dependency of the wavelengths. A more

thorough analysis of the dispersion error of the discretized free-surface flow problem is given in Chapter 5.

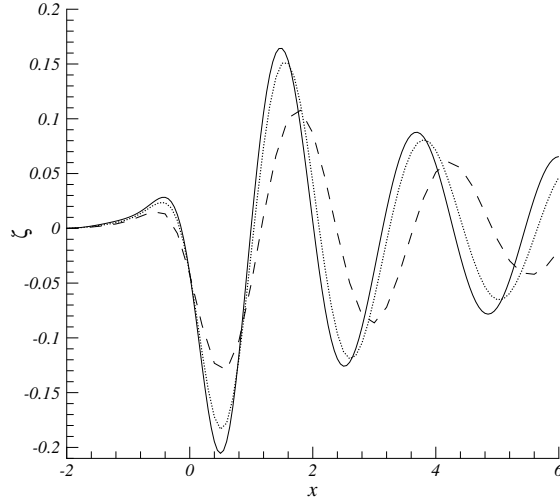


Figure 2.13: Wave elevation in the plane  $y = 0$  for the Gaussian pressure perturbation with  $P = 0.05$ ; on  $\hat{\Omega}_{2h}$  (dashed),  $\hat{\Omega}_h$  (dotted), and  $\hat{\Omega}_{\frac{h}{2}}$  (solid).

#### 2.4.5 Higher-order discretization of quasi free-surface boundary condition

As a next step, we replace the  $\mathcal{O}(h^2)$  upwind discretization of  $\nabla\varphi$  in (2.22) by an  $\mathcal{O}(h^3)$  upwind scheme. Then, the modified equation reads

$$u \frac{\partial p}{\partial x} = -u \frac{h^3}{12} \frac{\partial^4 p}{\partial x^4} + \mathcal{O}(h^4), \quad (2.27)$$

showing no dispersion error, but an  $\mathcal{O}(h^3)$  dissipation error, which is responsible for a decrease of the wave elevation. Note that the current  $\mathcal{O}(h^3)$  dissipation error is smaller than that of (2.25). A comparison of the solution obtained with the two schemes, after a single free-surface update, is shown in Figure 2.14, together with the results obtained through the potential-flow method described in [60]. As expected, the  $\mathcal{O}(h^3)$  scheme gives slightly higher waves. But it also yields a slightly less fast convergence of both the inner and outer iteration than the  $\mathcal{O}(h^2)$  scheme (compare Figure 2.15's left graph with Figure 2.8's left graph, and Figure 2.15's right graph with Figure 2.9). In the remainder of this thesis we do not use the  $\mathcal{O}(h^3)$  discretization of the quasi free-surface boundary condition.

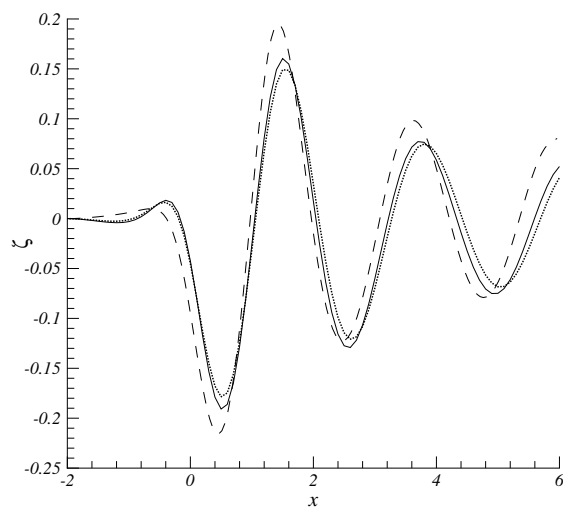


Figure 2.14: Wave elevation in the plane  $y = 0$  for the Gaussian pressure perturbation with  $P = 0.05$ ; for the Navier-Stokes method on  $\hat{\Omega}_h$ ,  $\mathcal{O}(h^3)$  scheme (solid) and  $\mathcal{O}(h^2)$  scheme (dotted); and for the potential-flow method from [60] (dashed).

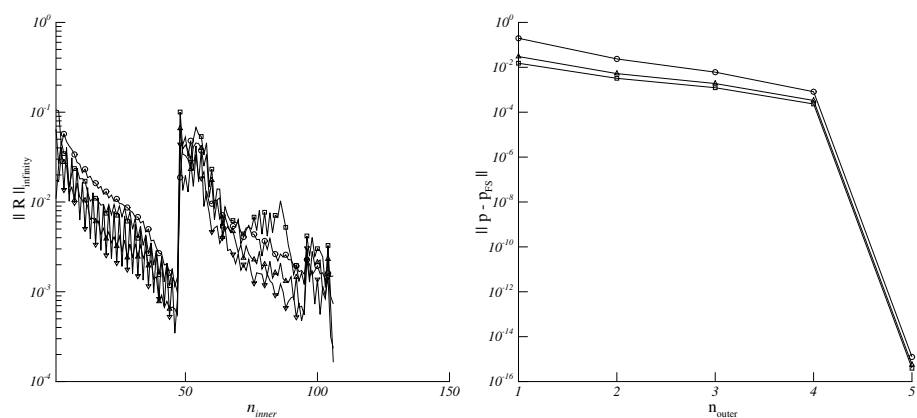


Figure 2.15: Convergence histories for the Gaussian pressure perturbation with  $P = 0.05$ , on  $\hat{\Omega}_h$ , with  $\mathcal{O}(h^3)$  discretization of the quasi free-surface boundary condition. *Left*: of the inner-iteration process,  $\mathbf{R}$  is the residual, measured on  $\hat{\Omega}_h$ , of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ), the  $y$ -momentum equation ( $\circ$ ), and the  $z$ -momentum equation ( $\nabla$ ), only every fourth marker is shown. *Right*: of the free-surface pressure defect, measured on  $\hat{\Omega}_h$ ; in  $L_1$ -norm ( $\square$ ),  $L_2$ -norm ( $\Delta$ ), and  $L_\infty$ -norm ( $\circ$ ).

### 2.4.6 Case with increased amplitude

The present numerical study concerns the computation of the wave pattern on  $\hat{\Omega}_h$ , for a stronger imposed pressure perturbation. The amplitude of the perturbation is increased from  $P = 0.05$  to  $P = 0.2$ , leaving the other parameters unaltered. The convergence history of the inner iteration is shown in the left graph of Figure 2.16. The computed wave elevation, in the plane of symmetry, is shown in the right graph of Figure 2.16. This figure shows that the deepest trough has fallen off to approximately  $\zeta = -0.88$ , instead of  $\zeta = -0.18$  for the  $P = 0.05$  case, which indicates that the wave system behaves nonlinearly.

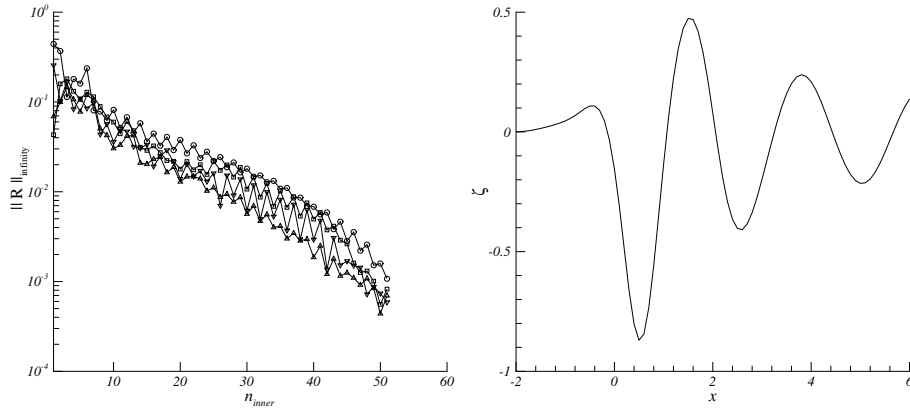


Figure 2.16: Computational results for the Gaussian pressure perturbation with  $P = 0.2$ , on  $\hat{\Omega}_h$ . *Left*: convergence history of the inner-iteration process,  $\mathbf{R}$  is the residual of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ), the  $y$ -momentum equation ( $\circ$ ), and the  $z$ -momentum equation ( $\nabla$ ), only every fourth marker is shown. *Right*: wave elevation in the plane  $y = 0$ .

## 2.5 Modified algorithm and numerical results

### 2.5.1 The modified algorithm

The free-surface algorithm as described in Section 2.3.1 is computationally efficient but the efficiency may be further improved. Alternately, the existing algorithm (i) completely solves the Navier-Stokes equations with the free surface frozen, and (ii) updates the free surface with the Navier-Stokes flow frozen. Complete solution of the Navier-Stokes flow per free-surface update may be unnecessary. Here we investigate the alternative approach of updating the free surface after each relaxation sweep in the Navier-Stokes solver. The modified algorithm still alternates between the bulk-flow update and the free-surface update (the complexity of a monolithic solution of Navier-Stokes flow and free surface is still avoided), but the frequency of alternating is much higher than in the original algorithm.

Summarizing, the modified free-surface algorithm reads:

$n := 0$

I.  $n := n + 1$

For a given boundary  $\Gamma_{\text{FS}}$ , perform a single relaxation sweep (i.e., symmetric for  $n$  odd, downwind only for  $n$  even) to

$$\begin{aligned} \mathcal{NS}(\mathbf{u}(\mathbf{x}), \varphi(\mathbf{x})) &= 0, \quad \forall \mathbf{x} \in \Omega, \\ \mathcal{B}(\mathbf{u}(\mathbf{x}), \varphi(\mathbf{x})) &= 0, \quad \forall \mathbf{x} \in \Gamma_0, \\ \left. \begin{aligned} \mathbf{t}^{(\alpha)} \cdot \boldsymbol{\tau}(\mathbf{u}) \cdot \mathbf{n} &= 0, \quad \alpha = 1, 2 \\ \mathbf{u} \cdot \nabla \varphi - \text{Fr}^{-2} \mathbf{u} \cdot \mathbf{e}_z &= \mathbf{u} \cdot \nabla p_{\text{FS}} \end{aligned} \right\}, \quad \forall \mathbf{x} \in \Gamma_{\text{FS}}. \end{aligned}$$

II. Use the new iterate  $(\mathbf{u}(\mathbf{x}), \varphi(\mathbf{x}))^T$  of I to obtain a new approximation of  $\Gamma_{\text{FS}}$  according to

$$\{(x, y, z := \text{Fr}^2(\varphi(\mathbf{x}) - p_{\text{FS}}(\mathbf{x})) : \forall \mathbf{x} \in \Gamma_{\text{FS}}\},$$

If  $\|\mathbf{R}\|_{\infty} < \epsilon_{\text{NS}}$  and  $\|p - p_{\text{FS}}\| < \epsilon_{\text{FS}}$  then stop, else goto I.

### 2.5.2 First results

The modified algorithm described in the previous section is now applied to compute the gravity waves for the  $P = 0.05$  case, leaving the other parameters unaltered as well. The computation is performed on the grid  $\hat{\Omega}_h$ . As a measure of convergence we will still monitor the residuals of the governing Navier-Stokes equations and the pressure defect at the free surface (Figure 2.17).

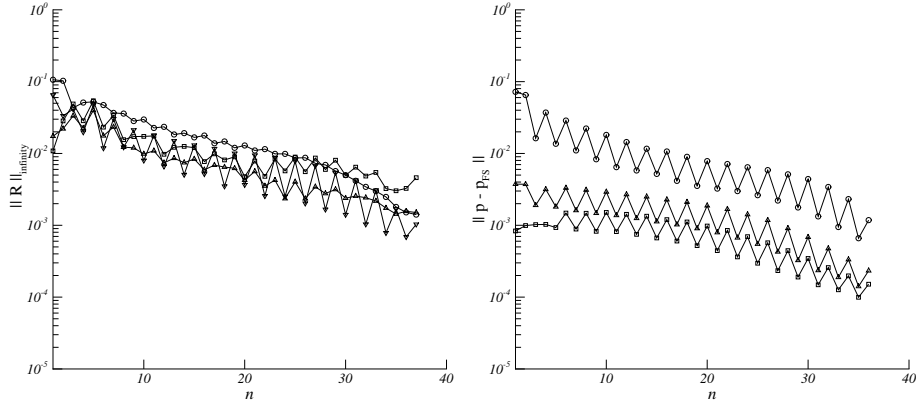


Figure 2.17: Convergence histories for the Gaussian pressure perturbation with  $P = 0.05$ , computed with the modified algorithm, on  $\hat{\Omega}_h$ . *left*: of  $\mathbf{R}$ , the residual of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ), the  $y$ -momentum equation ( $\circ$ ), and the  $z$ -momentum equation ( $\nabla$ ). *Right*: of the free-surface pressure defect; measured in  $L_1$ -norm ( $\square$ ),  $L_2$ -norm ( $\Delta$ ), and  $L_\infty$ -norm ( $\circ$ ).

Remarkable in Figure 2.17 is the regular oscillation in the convergence history of the free-surface residual  $\|p - p_{FS}\|$ . This oscillatory behavior is explained by the fact that the Navier-Stokes solver alternates between symmetric and one-sided (down-wind) relaxation sweeps, an experimentally found optimal relaxation strategy from which we do not deviate. Performing a free-surface update only after each symmetric relaxation sweep (but maintaining the alternate symmetric – anti-symmetric relaxation) removes this oscillatory behaviour, as can be seen in Figure 2.18.

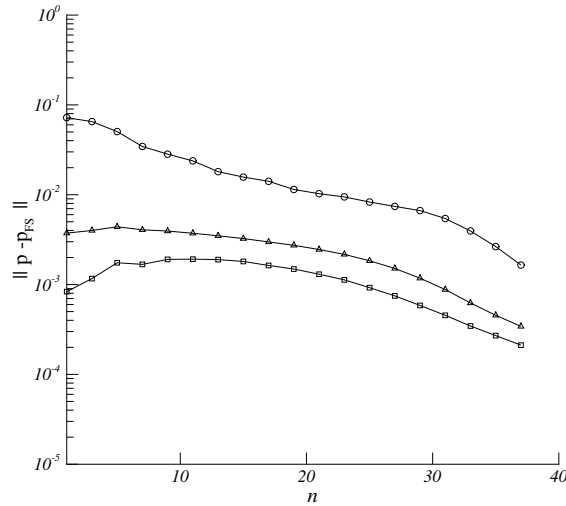


Figure 2.18: Convergence history of the pressure defect measured in  $L_1$ -norm ( $\square$ ),  $L_2$ -norm ( $\Delta$ ) and  $L_\infty$ -norm ( $\circ$ ); for the Gaussian pressure perturbation with  $P = 0.05$ , computed with the modified algorithm on  $\hat{\Omega}_h$  and with free-surface updates after each symmetric sweep only.

Comparing the total amounts of computational work put into the original algorithm and the modified algorithm (the left graphs in Figures 2.4 and 2.17, respectively), it appears that the modified algorithm is about six times more efficient than the original, a significant improvement. The convergence history of the wave elevation in the symmetry plane is depicted in Figure 2.19.



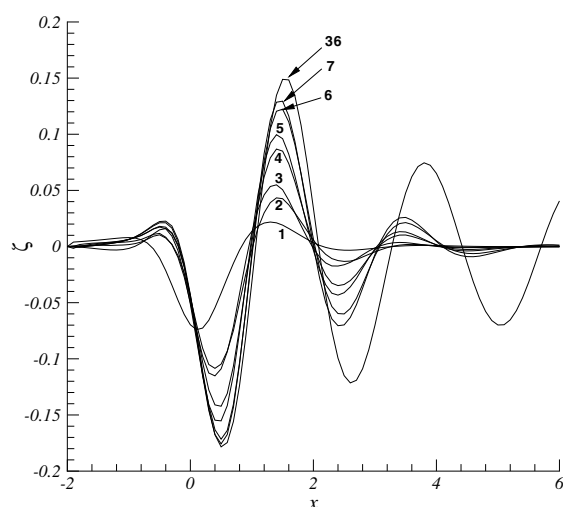


Figure 2.19: Convergence history of the wave elevation in the plane  $y = 0$  for the Gaussian pressure perturbation with  $P = 0.05$ , computed with the modified algorithm, on  $\hat{\Omega}_n$ , numbers denote the iteration counter  $n$ .

As expected, in comparison to the original algorithm, the modified algorithm needs more free-surface updates, but the computational work between two consecutive free-surface updates is of course very much less. Figure 2.19 nicely shows how the wave train literally unfolds itself; the first iterate (marked in Figure 2.19) shows a significant (though still far from converged) trough, a still very modest wave crest downstream and practically no waves further downstream. In the following iterates, besides increasingly larger wave amplitudes, also an increasingly longer wave train arises. Also note the differences in the free-surface changes between the symmetric ( $n = 1, 3, 5, \dots$ ) and one-sided ( $n = 2, 4, 6, \dots$ ) relaxation sweeps. Performing a free-surface update after each symmetric sweep only (still maintaining the original alternate symmetric – anti-symmetric relaxation strategy) yields the wave elevations denoted by  $1, 3, 5, \dots$  in Figure 2.20. This also suggests a further improvement in efficiency if the free surface is only updated after the symmetric relaxation sweep.

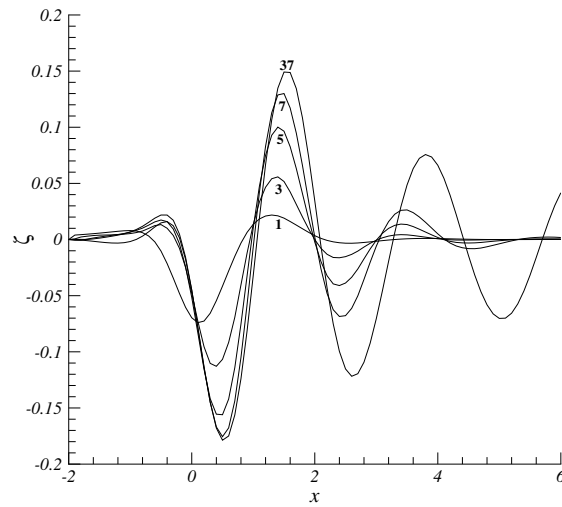


Figure 2.20: Convergence history of the wave elevation in the plane  $y = 0$  for the Gaussian pressure perturbation with  $P = 0.05$ , computed with the modified algorithm performing a free-surface update after **symmetric sweeps only**, on  $\hat{\Omega}_h$ , numbers denote the iteration counter  $n$ .

A consequence of this more evolutionary convergence of the free surface is that the Navier-Stokes method starts each relaxation on a grid that is less different from the preceding grid as compared to the original algorithm. As a consequence, *besides more efficient*, the modified algorithm is *also more robust*. The greater robustness manifests itself in, e.g., the fact that higher wave elevations can be obtained (to be shown in the next section).

### 2.5.3 Further results

To give an impression of the good robustness of the modified algorithm, in Figures 2.21–2.26 we present free-surface results for computations with successively  $P = 0.2, 0.3$  and  $0.4$ . The larger  $P$ , the more iterations are required, but the algorithm remains to be convergent. In the  $P = 0.4$ -case, a relative water height of more than 80% is obtained (Figure 2.26).

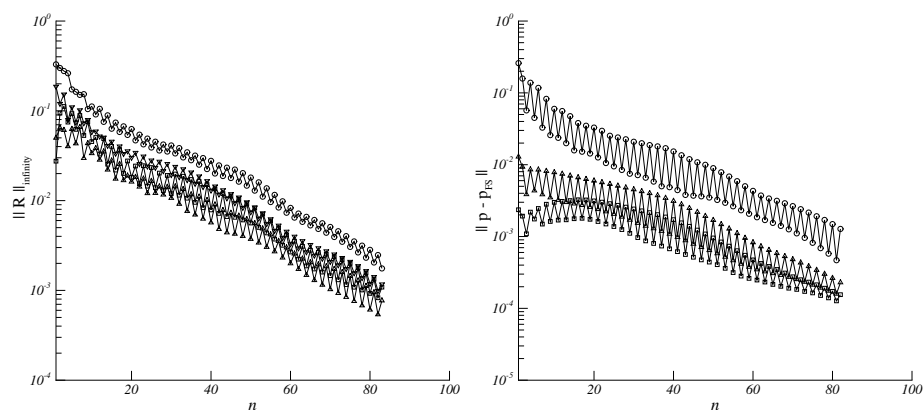


Figure 2.21: Convergence histories for the Gaussian pressure perturbation with  $P = 0.2$ , computed with the modified algorithm, on  $\hat{\Omega}_h$ . *Left:* of  $\mathbf{R}$ , the residual of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ), the  $y$ -momentum equation ( $\circ$ ), and the  $z$ -momentum equation ( $\nabla$ ). *Right:* of the free-surface pressure defect; measured in  $L_1$ -norm ( $\square$ ),  $L_2$ -norm ( $\Delta$ ), and  $L_\infty$ -norm ( $\circ$ ).

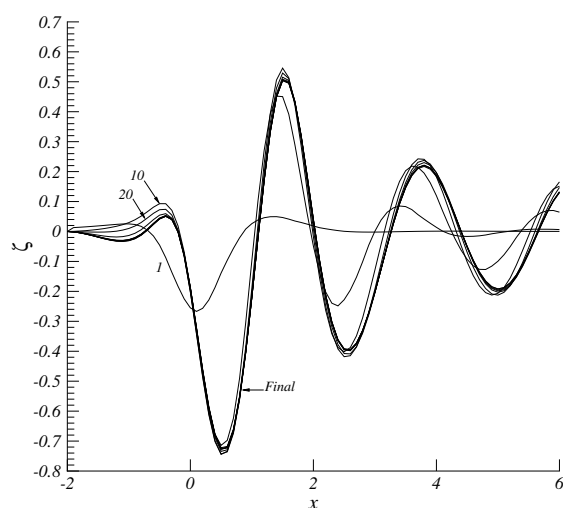


Figure 2.22: Convergence history of the wave elevation in the plane  $y = 0$  for the Gaussian pressure perturbation with  $P = 0.2$ , computed with the modified algorithm, on  $\hat{\Omega}_h$ , numbers denote the iteration counter, final iteration number  $n$  is 82.

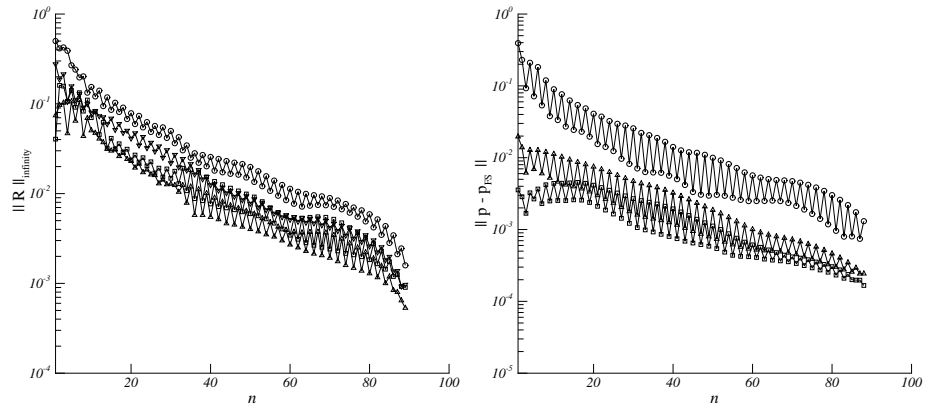


Figure 2.23: Convergence histories for the Gaussian pressure perturbation with  $P = 0.3$ , computed with the modified algorithm, on  $\hat{\Omega}_h$ . *Left:* of  $\mathbf{R}$ , the residual of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ), the  $y$ -momentum equation ( $\circ$ ), and the  $z$ -momentum equation ( $\nabla$ ). *Right:* of the free-surface pressure defect; measured in  $L_1$ -norm ( $\square$ ),  $L_2$ -norm ( $\Delta$ ), and  $L_\infty$ -norm ( $\circ$ ).

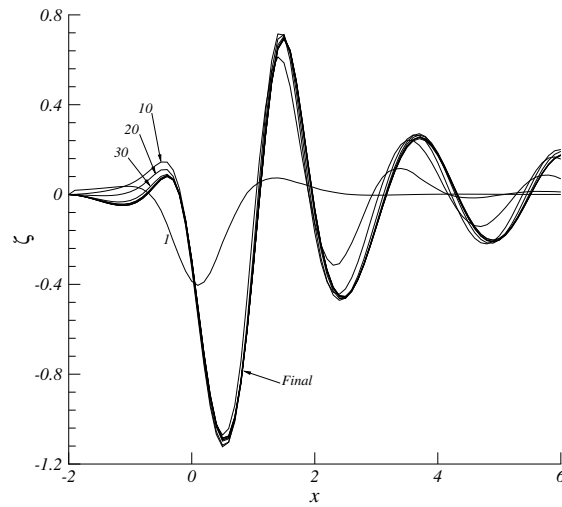


Figure 2.24: Convergence history of the wave elevation in the plane  $y = 0$  for the Gaussian pressure perturbation with  $P = 0.3$ , computed with the modified algorithm, on  $\hat{\Omega}_h$  numbers denote the iteration counter, final iteration number  $n$  is 88.

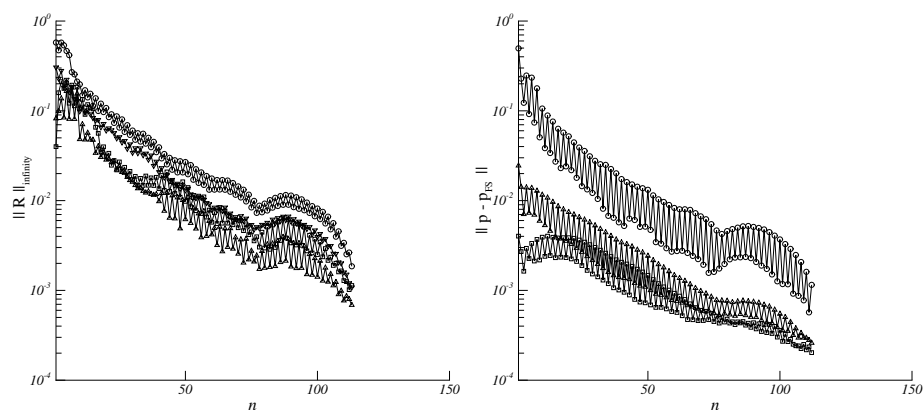


Figure 2.25: Convergence histories for the Gaussian pressure perturbation with  $P = 0.4$ , computed with the modified algorithm, on  $\hat{\Omega}_h$ . *Left:* of  $\mathbf{R}$ , the residual of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ), the  $y$ -momentum equation ( $\circ$ ), and the  $z$ -momentum equation ( $\nabla$ ). *Right:* of the free-surface pressure defect; measured in  $L_1$ -norm ( $\square$ ),  $L_2$ -norm ( $\Delta$ ), and  $L_\infty$ -norm ( $\circ$ ).

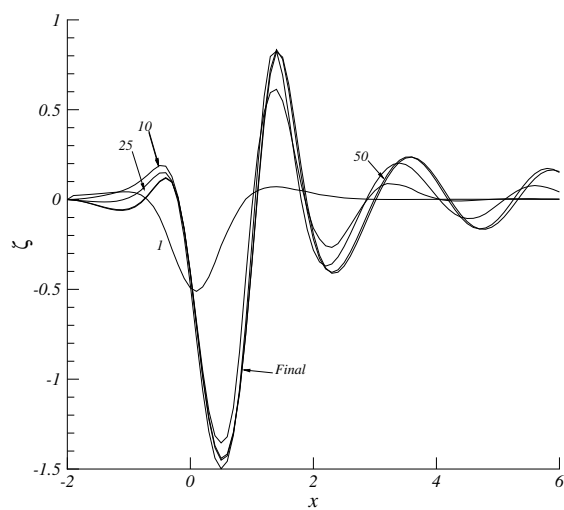


Figure 2.26: Convergence history of the wave elevation in the plane  $y = 0$  for the Gaussian pressure perturbation with  $P = 0.4$ , computed with the modified algorithm, on  $\hat{\Omega}_h$  numbers denote the iteration counter, final iteration number  $n$  is 113.

## 2.6 Conclusions

Solution of the steady, free-surface Navier-Stokes equations through a time-stepping approach is known to be inefficient, particularly in 2D but also in 3D (convergence with  $\frac{1}{\sqrt{t}}$  and  $\frac{1}{t}$ , respectively). For the 2D, free-surface Navier-Stokes equations, Van Brummelen et al. have proposed a non-monolithic free-surface algorithm that does not follow a time-stepping approach. In the present chapter we have extended this algorithm to 3D and have applied it to flow problems with increasing nonlinearity of the wave system.

Our results show that, for convergence purposes, it makes sense (i) to take the computational domain sufficiently large and (ii) to discretize the quasi free-surface boundary condition only first-order accurate in the far field. (In this way, unperturbed far-field boundary conditions can be imposed.) The results also show that the exclusion of the wave-dissipation zone in the calculation of the norm of the free-surface pressure defect is beneficial to the free-surface convergence rate. This approach is perfectly acceptable because the wave-dissipation zone is outside our domain of interest.

The results also reveal that reinitializing the preconditioner after each free-surface update has an adverse effect on the computational efficiency of the solution process of the Navier-Stokes subproblem. Not reinitializing the preconditioner after each free-surface update improves the convergence of the outer-iteration process.

The considered 3D free-surface algorithm appears to quickly yield the proper 3D wave physics. The free-surface pressure defect seems to converge rather grid-independently. For linear and mildly nonlinear wave systems, free-surface iteration may not even be necessary though; only a single free-surface update may be sufficient for finding the wave pattern to within engineering accuracy.

An important result of the chapter is the proposed reduction of the inner iteration to a single work unit only (a single relaxation sweep). This reduction yields a significant improvement in both efficiency and robustness. This modified version of the free-surface algorithm is expected to be useful to ship hydrodynamics in an industrial context. It is further applied in Chapter 4.

## CHAPTER 3

# Evaluation of Nonuniform Meshes for 3D Navier-Stokes Flow

### 3.1 Introduction

In the previous chapter, through numerical experiments, we investigated the convergence and accuracy of our 3D free-surface iteration method. In Chapter 1 we stated that our main goal is to develop a numerical method for the computation of the wave pattern generated by a ship hull. In numerical ship hydrodynamics *two* key features are predominant: (i) the free surface and (ii) the boundary layer. Both features possess length scales which are very dissimilar: the free surface length scale  $\ell$ , which is typically the ship's length, and the boundary-layer thickness  $\delta$ , for which it holds:  $\delta \ll \ell$ . The mutual influence of these two length scales cannot be neglected if an accurate approximation of the flow field and any derived quantity, such as, e.g., the total drag of the ship, is to be computed with a numerical method. To properly account for both free-surface and boundary-layer features the 3D, incompressible Reynolds-averaged Navier-Stokes equations have to be solved in conjunction with the free-surface conditions. This is a non-trivial matter, particularly at the intersection of the free surface with the ship hull, due to the incompatibility of the kinematic free-surface condition, imposed on the free surface, and the no-slip boundary condition, imposed on the hull. This so-called 'contact-line' problem can be seen as follows. Substitution of the no-slip condition  $\mathbf{u} = \mathbf{0}$  into (2.8) yields

$$\eta_t = 0, \quad (3.1)$$

hence

$$\eta(\mathbf{x}, t) = \eta(\mathbf{x}, 0), \quad \forall \mathbf{x} \in \Gamma_{ns}, \quad t \geq 0, \quad (3.2)$$

where

$$\Gamma_{ns} = \{\mathbf{x} \in \partial\Omega : \mathbf{u}(\mathbf{x}) = \mathbf{0}\}, \quad (3.3)$$

where *ns* refers to no-slip. This means that along the intersection line the free surface does not move from its initial position. Hence the waterline cannot be determined with any numerical method based on this formulation of the free surface. A common solution often encountered in numerical algorithms uses an extrapolation

of the local wave height near the hull, see, e.g., [16, 73]. This approach is justified by the argument that the mesh width near the ship hull, in normal direction, is small so that extrapolation can be done accurately. We will show, in a later section, that the free-surface iteration method proposed in this thesis does not suffer from the contact-line problem.

Since we consider high Reynolds-number flows the corresponding boundary layers are very thin. To accurately resolve the boundary layer on a uniform mesh, the total number of grid points normal to the flow direction is proportional to  $\text{Re}^{\frac{1}{2}}$  in case of a laminar boundary layer. This results in an unreasonably large number of nodes, especially in  $\mathbb{R}^d$ ,  $d > 1$ .

The abovementioned situation strongly advocates the necessity of non-uniform meshes. There are two main variants of this approach. One may choose an *a-priori* refined mesh, based on a-priori knowledge of the solution, i.e., location and some measure of the thickness of the boundary layer. Or one may compute some approximation on a given mesh, and then use this information to adapt the mesh *a-posteriori*. In the category of a-priori refined meshes theoretical results on (i) parameter-uniform convergence of the iterative method and on (ii) accuracy of the approximate solution have been obtained for singularly perturbed differential equations in  $\mathbb{R}^d$ . See Chapter 4 in [79] for an introduction to the subject and [24, 63] for a more extensive overview. This theory has led to the simplest non-uniform mesh, namely the piecewise-uniform mesh which is fitted to the boundary layer. Such a mesh is composed of two uniform submeshes: a fine submesh in the boundary layer and a coarse submesh outside the boundary layer. The transition between the fine and the coarse submesh is determined by specifying the location of the transition, denoted by  $\sigma$ . In [24] it is argued that  $\sigma = \sigma(\varepsilon, N)$ , where  $\varepsilon$  is the singular perturbation parameter, say  $\varepsilon = \text{Re}^{-1}$ , and  $N$  the total number of points in the direction normal to the flow. The intuitive argument for the choice of  $\sigma$  is that it should be chosen such that  $\sigma > \delta(N)$  where  $\delta(N)$  is some discrete measure of the boundary layer thickness. For a known boundary-layer function  $f(x)$ , e.g.,  $f(x) = e^{-\frac{x}{\varepsilon}}$ , the computational width  $\sigma$  of the boundary layer may be required to satisfy the condition that for  $x > \sigma$  the function values  $f(x)$  are smaller than the discretization error. Assume that, in formula, this may be written as

$$\sup_{x > \sigma} |f(x)| < N^{-p}, \quad (3.4)$$

where  $p$  is the order of accuracy of the discretization. From (3.4), for  $f(x) = e^{-\frac{x}{\varepsilon}}$  it follows

$$\sigma = p\varepsilon \ln N. \quad (3.5)$$

The type of mesh for  $p = 1$  and with the coarse and the fine submesh each containing  $\frac{N}{2}$  mesh points, is known as a *Shishkin mesh*. The advantage of a Shishkin mesh is its simplicity. However, this mesh has as major drawback that a-priori knowledge about the thickness of the layer structure is imperative. In general, e.g., for turbulent boundary layers, this knowledge is not available a-priori.

A frequently encountered second type of non-uniform mesh is the *exponentially stretched mesh*. Here the transition from the fine to the coarse mesh is more gradual



depending on the form of the mesh generating function. Various mesh generating functions can be conceived. An example of an exponentially stretched mesh is the *Bakhvalov mesh*, see, e.g., [63]. Assuming a boundary-layer function of the form  $f(x) = e^{-\beta \frac{x}{\varepsilon}}$ ,  $\beta > 0$ , Bakhvalov's idea is to use an equidistant  $f(x)$ -grid, and then to map this grid back onto the  $x$ -axis by means of the boundary-layer function, see Figure 3.1. That is, the grid points  $x_i$  near  $x = 0$  are defined by

$$x_i = -\frac{\varepsilon}{\beta} \ln\left(1 - \frac{i}{N}\right). \quad (3.6)$$

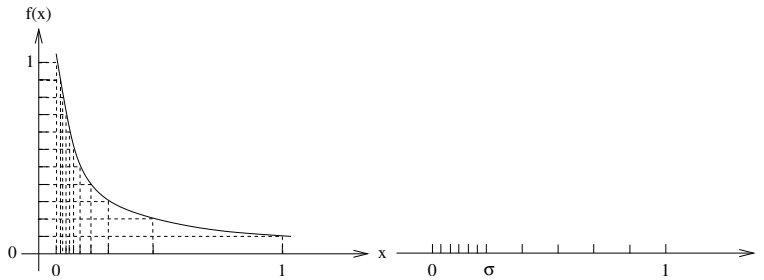


Figure 3.1: Schematic representation of the Bakhvalov mesh (*left figure*) and the Shishkin mesh (*right figure*).

An advantage of the piecewise-uniform mesh – which in fact motivates this research – is that its generation is so simple, and therefore also its a-posteriori adaptation. Hence, our primary focus, in this chapter, is on the viability of the piecewise-uniform mesh for engineering type flow problems, particularly flow problems which involve a free surface. As a model problem, we compute the flow along a partially submerged flat plate in a uniform horizontal flow with a pressure perturbation imposed at the free surface, a perturbation of the form (2.21). This model problem contains the relevant flow features which would also be encountered in an actual ship hydrodynamics problem, i.e., a boundary layer, stagnation points and a free surface which intersects the boundary layer, but it lacks the geometric complexity. To resolve the boundary layer, both types of meshes, i.e., the piecewise uniform and the exponentially-stretched one, are considered.

The numerical results of this chapter can be divided into three parts. In the first part any free-surface effect is neglected and the emphasis lies on the computation of the boundary-layer flow along the flat plate, on the piecewise-uniform and the exponentially-stretched mesh. Here we assume that the flow is laminar and we consider a single value of  $Re$  only. This flow problem is known as Prandtl's problem and has a semi-analytical solution, known as the Blasius solution, see, e.g., [64]. This solution will serve as a reference solution at the center of the plate. The second part will focus on the computation of the same laminar boundary layer for varying Reynolds numbers ( $10^6 - 16 \times 10^6$ ), again neglecting the free-surface flow. In the third part the free-surface flow is no longer ignored. Here the aim is to compute the boundary layer as well as the shape of the free surface. In this part we assume

the boundary layer to be fully turbulent. The turbulence is accounted for through the use of an algebraic turbulence model for the eddy viscosity  $\nu_t$ . The free-surface iteration method employed here corresponds to the method described in Section 2.3.1. In adjustments of the free-surface boundary, the underlying mesh points have to be redistributed to keep the mesh smooth. It is our idea that this free-surface mesh adaptation may be nicely combined with an adaptation of the mesh to fit the computed boundary layer. For the Shishkin mesh, this only requires some a-posteriori measure of the computational width of the boundary layer to determine the transition parameter  $\sigma$ .

### 3.2 Computational method

The viscous free-surface flow problem studied in this chapter is identical to the one described in the previous chapter, except for the presence of the flat plate. The boundary-value problem which has to be solved and the numerical methods which are applied to solve the discrete boundary-value problem correspond to the descriptions given in Sections 2.2 and 2.3, respectively. In this section we will restrict ourselves to describing the deviations only.

Let  $\Omega \subset \mathbb{R}^3$  denote the computational domain bounded by  $\partial\Omega = \Gamma_0 \cup \Gamma_{\text{FS}}$ , where  $\Gamma_{\text{FS}}$  denotes the free surface and  $\Gamma_0$  the remaining (fixed) part of  $\partial\Omega$ . The computational domain is bounded by

$$\Omega = \{(\mathbf{x}) : [-1, 3] \times [0, 4] \times [-4, \eta]\}, \quad (3.7)$$

where  $\eta : [-1, 3] \times [0, 4] \rightarrow \mathbb{R}$  represents the wave-height function. On  $\Gamma_0$ , the boundary conditions (2.15) – (2.19) are imposed. However, in this chapter the internal boundary  $y = 0$  is decomposed into a symmetry plane and the flat plate, defined by

$$\Gamma_{ns} = \{(\mathbf{x}) : -\frac{1}{2} \leq x \leq \frac{1}{2}, y = 0, -4 \leq z \leq \eta\}, \quad (3.8)$$

on which the ‘no-slip’ boundary condition is imposed, i.e.,

$$u(\mathbf{x}) = 0, \quad v(\mathbf{x}) = 0, \quad w(\mathbf{x}) = 0, \quad \forall \mathbf{x} \in \Gamma_{ns}. \quad (3.9)$$

In Sections 3.3.1 and 3.3.2 the free-surface flow is still ignored by replacing the free-surface boundary conditions (2.20) and (2.22), imposed at  $z = \eta$ , by (2.19) imposed at  $z = 0$ . This effectively reduces the 3D flow problem to a 2D flow problem in the  $(x, y)$ -plane.

As initial solution for the iteration method the uniform-flow solution is used.

### 3.3 Numerical results

In Section 3.3.1 and Section 3.3.2, a comparison is made of the laminar boundary layer computed on a piecewise-uniform mesh and on an exponentially stretched mesh, where in Section 3.3.1 a variation of the mesh spacing is considered at fixed Reynolds number and in Section 3.3.2 a variation of the Reynolds number at fixed

mesh size. In both sections the free-surface flow is not considered yet. Section 3.3.3 constitutes the same problem as before but now the free-surface flow is included and the boundary layer is now assumed to be turbulent.

### 3.3.1 Laminar boundary layer

The Reynolds number is set at  $\text{Re} = 10^6$  and the flow is assumed to be laminar. It is known that this problem does not have an analytical solution in closed form. We use the semi-analytical first-order solution of Blasius as a reference for the boundary-layer velocity profiles, see [64]. Note that this solution is not valid near the two singular points, i.e., the leading and trailing edge of the plate. It has been shown that in a region around the leading edge of the plate, a region with size  $\mathcal{O}(\text{Re}^{-1})$ , the full Navier-Stokes equations have to be solved, see e.g. [78]. Near the trailing edge the situation is even more complex. Here, according to triple-deck theory, several regions with sizes depending on  $\text{Re}^{-\alpha}$ ,  $\alpha \in \mathbb{Q}^+$ , can be identified. In a region with radius  $\mathcal{O}(\text{Re}^{-3/4})$ , and centered around the trailing edge, the full Navier-Stokes equations have to be solved. This region will be smaller than the smallest mesh width on our finest mesh. Hence, the reduction of our viscous operator, i.e.  $\tau_{,x}^{\alpha x} = 0$ , is still warranted.

#### Piecewise-uniform mesh

First, we construct the piecewise-uniform fitted mesh. Because the computational domain is rectangular, the mesh  $\Omega^N$  is the tensor product of three 1D-meshes, i.e.  $\Omega^N = \Omega^{N_x} \times \Omega^{N_y} \times \Omega^{N_z}$ , where  $N_x$ ,  $N_y$  and  $N_z$  denote the number of nodes in the  $x$ -,  $y$ - and  $z$ -direction, respectively. Since the boundary layer is in the  $y$ -direction, only  $\Omega^{N_y}$  has a piecewise-uniform mesh fitted on the total interval  $y \in [0, 4]$  such that each of the subintervals  $[0, \sigma]$  and  $[\sigma, 4]$  has  $\frac{N_y}{2}$  uniformly spaced nodes. So, the mesh widths for the fine and coarse submesh are

$$h_y = \frac{2\sigma}{N_y} \quad \text{and} \quad H_y = \frac{2y_e}{N_y} - h_y, \quad \text{respectively, with } y_e = 4. \quad (3.10)$$

$\Omega^{N_x}$  and  $\Omega^{N_z}$  are uniform meshes. To completely define the mesh it remains to specify the value of  $\sigma$ . In the foregoing section for the case of an a-priori known exponential boundary-layer function  $e^{-\frac{x}{\varepsilon}}$ , we derived  $\sigma = p\varepsilon \ln N$ . Since in the case of Prandtl's problem the boundary-layer function is not known in analytical form, but since we do know that – as in the  $e^{-\frac{x}{\varepsilon}}$  model problem – the dependence of  $\varepsilon$  is dominant over  $N$ , we simply neglect the dependence of  $\sigma$  on  $N$  and assume that  $\sigma = \sigma(\sqrt{\varepsilon})$ , where  $\varepsilon = \text{Re}^{-1}$ . Here we choose  $\sigma = C\delta$ , where  $C$  is an arbitrarily chosen constant ( $C = 5$ ) and  $\delta$  is an estimate of the boundary-layer thickness at the trailing edge of the plate, which can be estimated with  $\delta = 5/\sqrt{\text{Re}}$ , see [64], resulting in

$$\sigma(\text{Re}) = 25/\sqrt{\text{Re}}. \quad (3.11)$$

The constant  $C$  is incorporated as to provide a safety margin so that the complete boundary layer is fitted in the part with the fine mesh. In the remainder of this chapter the piecewise-uniform mesh is denoted by  $\Omega^P$ .

### Exponentially-stretched mesh

The second type of mesh we consider here, is the exponentially graded type, which will be denoted by  $\Omega^E$ . This mesh is also the result of a tensor product of three 1D meshes. The non-uniform distribution is applied in the  $y$ -direction only. The mesh-generating function corresponds to the solution of the following boundary-value problem, find  $y(\zeta) : [0, 1] \rightarrow \mathbb{R}$ , such that it satisfies

$$y''(\zeta) + \alpha y'(\zeta) = 0, \quad (3.12)$$

$$y(0) = 0, \quad y(1) = 4, \quad (3.13)$$

where  $\alpha \in \mathbb{R}^-$  is the stretching parameter, which is defined below. The discrete  $y(j)$  coordinates of the mesh points correspond to the set

$$y(j) = \{y(j\Delta\zeta), j \in [0, N] \subset \mathbb{N}, \Delta\zeta = N^{-1}\}. \quad (3.14)$$

The mesh spacing in the  $x$ - and  $z$ -direction is identical to that for the piecewise-uniform mesh. The stretching parameter follows from the (nonlinear) equality

$$(y(2) - y(1))|_{\Omega^P} = (y(2) - y(1))|_{\Omega^E}. \quad (3.15)$$

The computations are performed on a family of three meshes with densities  $N, 2N, 4N$ , where  $N = 20$ . The corresponding stretching parameters  $\alpha$  are given in Table 3.1.

$\Omega^{E,N}$	$\Omega^{E,2N}$	$\Omega^{E,4N}$
6.40518	6.30506	6.25701

Table 3.1: Computed stretching parameters  $\alpha$ .

In the remainder of this chapter the numerical results are monitored at the center of the flat plate, unless mentioned otherwise. To be able to compare the computed solution with the Blasius solution the velocity components  $u$  and  $v$  are scaled, according to

$$u := \frac{u}{U_\infty}, \quad v := \frac{v}{U_\infty} \sqrt{s \text{Re}} \quad \text{and} \quad \eta := \frac{y}{\sqrt{s}} \sqrt{\text{Re}}, \quad (3.16)$$

and are evaluated at  $s = \frac{1}{2}$ , which corresponds with  $x = 0$ , and at  $z = -2$  (hence at the middle of the plate). The scaled velocity-component distributions, computed on  $\Omega^P$  and  $\Omega^E$ , are shown in Figures 3.2 and 3.3, respectively.

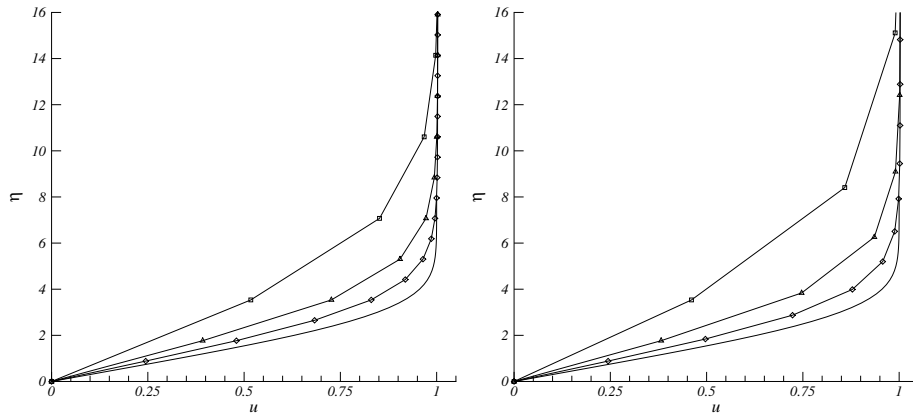


Figure 3.2:  $u$ -profile at center of plate, computed on  $\Omega^P$  (left figure) and  $\Omega^E$  (right figure) with mesh densities  $N(\square)$ ,  $2N(\Delta)$  and  $4N(\diamond)$ , and with the Blasius solution (solid). ( $N = 20$ ).

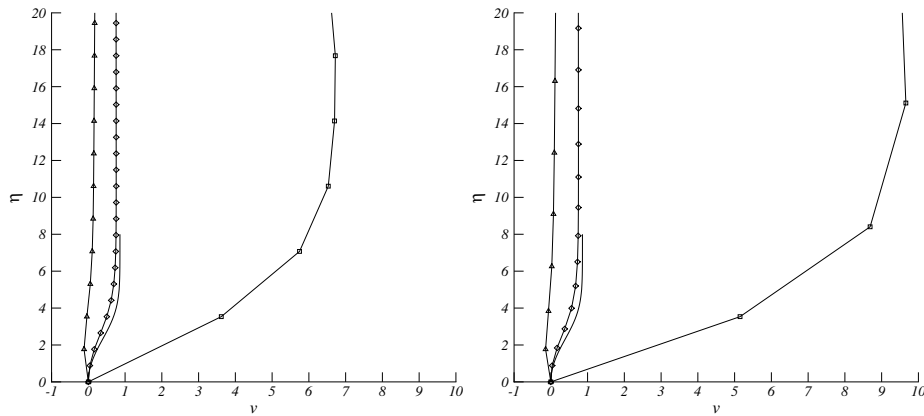


Figure 3.3:  $v$ -profile at center of plate, computed on  $\Omega^P$  (left figure) and  $\Omega^E$  (right figure) with mesh densities  $N(\square)$ ,  $2N(\Delta)$  and  $4N(\diamond)$ , and with the Blasius solution (solid). ( $N = 20$ ).

The mainstream velocity component  $u$  shows monotone convergence to the Blasius solution with decreasing mesh width for both types of meshes. However, Figure 3.3 reveals a non-monotone convergence, for both types of meshes. Note, for instance, the small negative value of  $v$  computed on both  $\Omega^{P,2N}$  and  $\Omega^{E,2N}$ . A clear comparison of the two distributions of the velocity components computed on  $\Omega^{P,4N}$  and  $\Omega^{E,4N}$  is made in Figure 3.4.

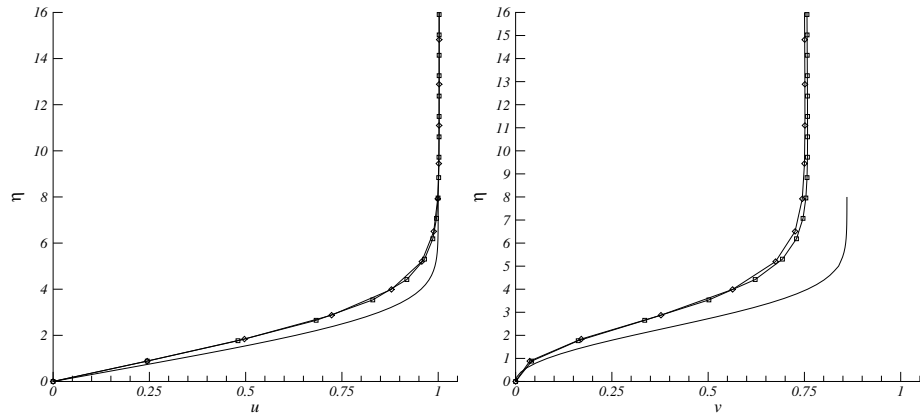


Figure 3.4:  $u$ -profile (left figure) and  $v$ -profile (right figure) at center of plate, computed on  $\Omega^{P,4N}$  ( $\square$ ) and  $\Omega^{E,4N}$  ( $\diamond$ ), and the Blasius solution (solid). ( $N = 20$ ).

From the computed velocity profiles as depicted in Figures 3.3 and 3.4 it can not be concluded which of the two types of grids performs best. However, the pressure reveals a significant difference, as can be seen in Figure 3.5.

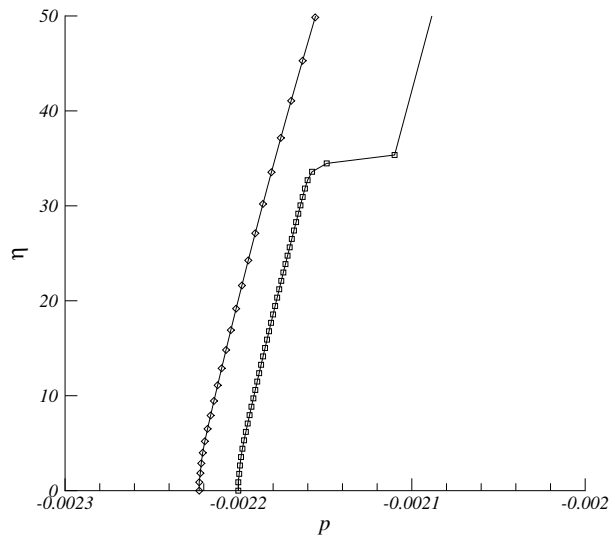


Figure 3.5: Pressure distribution at center of plate, computed on  $\Omega^{P,4N}$  ( $\square$ ) and  $\Omega^{E,4N}$  ( $\diamond$ ). ( $N = 20$ ).

The first-order approximation of the pressure field resulting from the classical boundary-layer theory is known to satisfy  $\frac{\partial p}{\partial \eta} = 0$ . For the Navier-Stokes computation this is not the case. Due to the fact that the center of our plate is not infinitely far from the leading edge, with as a consequence a still finite curvature

of streamlines, we find a still finite pressure gradient in  $\eta$ -direction. On  $\Omega^P$  the pressure shows a sudden erroneous jump at the transition from the coarse to the fine mesh. The slight difference in  $p(\eta = 0)$  for both types of meshes, as observed in Figure 3.5, is considered to be irrelevant.

The convergence of the residuals of the equations, for the computations on  $\Omega^P$  and  $\Omega^E$  for the three mesh densities, is shown in Figures 3.6, 3.7 and 3.8. The convergence is monitored using the  $L_\infty$ -norm. Employing norms which involve some kind of averaging, e.g.  $\|\cdot\|_1$ ,  $\|\cdot\|_2$  or the energy norm  $\|\cdot\|_{\text{energy}}$ , possesses the probability of not detecting the boundary layer. See [24] for a more elaborate discussion on proper norms for singularly perturbed problems.

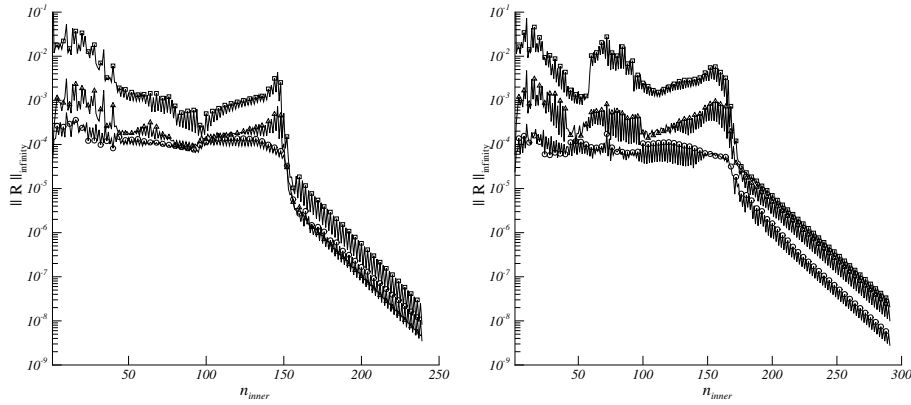


Figure 3.6: Convergence histories of the iteration process on  $\Omega^{P,N}$  (left figure) and  $\Omega^{E,N}$  (right figure).  $\mathbf{R}$  is the residual of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ) and the  $y$ -momentum equation ( $\circ$ ), only every fourth marker is shown. ( $N = 20$ ).

Let  $\mathbf{R} := (R_x, R_y, R_z, R_{\text{div}u})^T$  be the residual vector with as components the residuals of the governing equations, i.e., the momentum equations and the divergence of the velocity field. Of these four residuals, the  $R_z$ -component is of no significance since the boundary conditions specified prevent any dynamics in  $z$ -direction, and therefore this component is not plotted here. The 2D flow problem is restricted to the  $(x, y)$ -plane. Monitoring the location of  $\|\mathbf{R}\|_\infty$  on  $\Omega^{P,N}$  shows that the initially slow convergence of the  $y$ -momentum equation, is due to the sudden change of the mesh width at  $y = \sigma$ . The residuals  $\|R_y\|_\infty$ , for  $n_{\text{inner}} \geq 150$  occur near the external boundary,  $y = 4$ . The residuals  $\|R_x\|_\infty$  mostly occur in the boundary layer at the center of the plate or in the wake near the outflow boundary. On  $\Omega^{E,N}$  no specific area is observed for  $\|R_y\|_\infty$ .

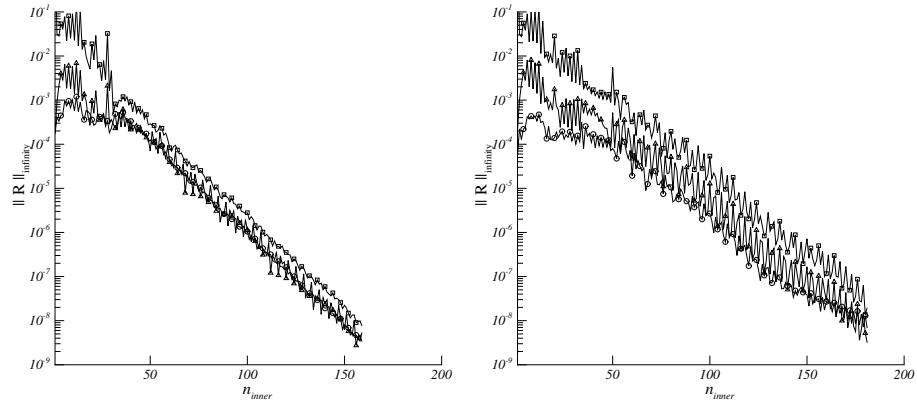


Figure 3.7: Convergence histories of the iteration process on  $\Omega^{P,2N}$  (left figure) and  $\Omega^{E,2N}$  (right figure).  $\mathbf{R}$  is the residual of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ) and the  $y$ -momentum equation ( $\circ$ ), only every fourth marker is shown. ( $N = 20$ ).

The locations of the residuals on  $\Omega^{P,2N}$  coincide with the sudden jump in the mesh width at  $y = \sigma$ , as was mostly the case with  $\Omega^{P,N}$ . On  $\Omega^{E,2N}$ , however, most residuals mainly occur in the boundary-layer region.

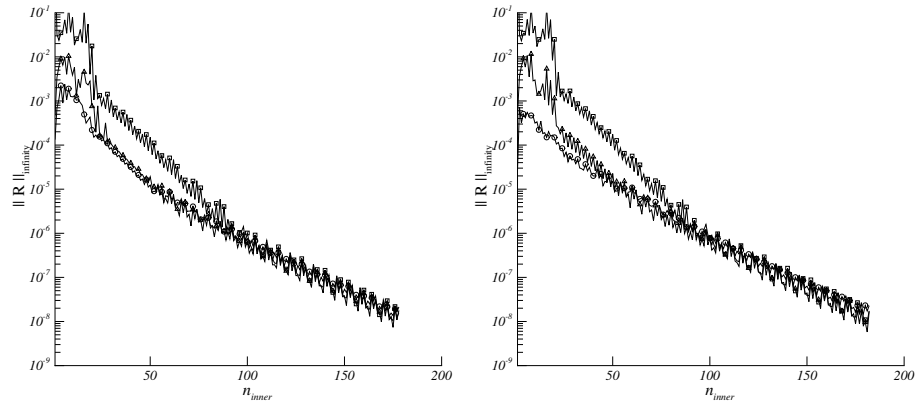


Figure 3.8: Convergence histories of the iteration process on  $\Omega^{P,4N}$  (left figure) and  $\Omega^{E,4N}$  (right figure).  $\mathbf{R}$  is the residual of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ) and the  $y$ -momentum equation ( $\circ$ ), only every fourth marker is shown. ( $N = 20$ ).

The Figures 3.6 – 3.8 show that the amount of work for the computation on the piecewise-uniform mesh is always slightly less than on the exponentially stretched mesh.



### 3.3.2 Variation of Reynolds number on piecewise-uniform mesh

In this section the variation of the Reynolds number is studied. Here we consider, in addition to  $\text{Re} = 10^6$ , as considered in the previous section, the Reynolds numbers  $\text{Re} = 4 \times 10^6$  and  $\text{Re} = 16 \times 10^6$ . All computations are performed on the piecewise-uniform mesh  $\Omega^{P,4N}$  with  $N = 20$ . The transition parameter  $\sigma$  is obtained from (3.11). The profiles of the velocity components  $u$  and  $v$ , at the center of the plate and *all* scaled for  $\text{Re} = 10^6$ , are shown in Figure 3.9.

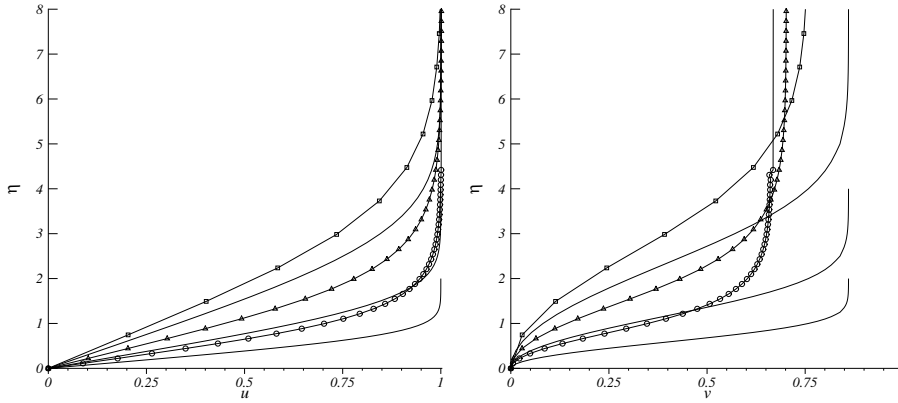


Figure 3.9:  $u$ -profile (left figure) and  $v$ -profile (right figure) at center of plate, computed on  $\Omega^{P,4N}$ , for  $\text{Re} = 10^6$  ( $\square$ ),  $\text{Re} = 4 \times 10^6$  ( $\Delta$ ) and  $\text{Re} = 16 \times 10^6$  ( $\circ$ ), and the Blasius solution (solid). The vertical scale corresponds with  $\text{Re} = 10^6$ . ( $N = 20$ ).

For each of the three Reynolds-number cases, it appears that, as expected, for the  $u$ -component, the difference between the computed solution and the Blasius solution increases toward the edge of the boundary layer, and then decreases to zero for further increasing  $\eta$ . From classical boundary-layer theory it follows that

$$\frac{\delta_4}{\delta_1} = \frac{\delta_{16}}{\delta_4} = \frac{1}{2}, \quad (3.17)$$

where  $\delta_k$  is the boundary-layer thickness belonging to  $\text{Re} = k \times 10^6$ . The numerical results plotted in the left graph of Figure 3.9 show that these fractions are

$$\frac{\delta_4}{\delta_1} = 0.68, \quad \text{and} \quad \frac{\delta_{16}}{\delta_4} = 0.63, \quad (3.18)$$

revealing a slowly converging Reynolds-number dependence, assuming that (3.17) is exact.

Furthermore, with increasing Reynolds number, for  $\eta \rightarrow \infty$ , the  $v$ -component shows an increasing deviation of the asymptotic value according to Blasius. From classical boundary-layer theory it follows that

$$v_\infty := \lim_{\eta \rightarrow \infty} v(\eta) = 0.8604, \quad (3.19)$$

where  $v$  is scaled according to (3.16). Table 3.2 shows the maximum values of  $v$ , corresponding with the three numerical curves in the right graph of Figure 3.9.

Re	$10^6$	$4 \times 10^6$	$16 \times 10^6$
$v_{\max}$	0.7555	0.7016	0.6692

Table 3.2: Computed maximum values of  $v$  for the various Reynolds numbers.

The observed deviation may have physical and numerical causes. The latter may be analyzed by deriving the leading-order terms of the truncation error, here denoted as  $\tau_h$ , resulting from the schemes described in Section 2.3.2. Assume, for the purpose of analysis, that the mesh width in  $x$ - and  $y$ -direction is uniform, i.e.,  $h_x = h_y = h$ . Then the truncation error is obtained via a Taylor-series expansion of the solution components  $u(x, y)$ ,  $v(x, y)$  and  $\varphi(x, y)$  around the point  $(x_i, y_j)$ . The dependence of the truncation error on the Reynolds number can be made explicit by redefining all variables through application of the scaling

$$x := \frac{x}{\ell}, \quad y := \frac{y}{\ell} \sqrt{\text{Re}}, \quad u := \frac{u}{U_\infty}, \quad v := \frac{v}{U_\infty} \sqrt{\text{Re}} \quad \text{and} \quad h := \frac{h}{\ell}. \quad (3.20)$$

This results in the following expressions for the leading-order terms of the truncation errors. For the continuity equation:

$$\tau_h = \frac{h^2}{3} \frac{\partial^3 u}{\partial x^3} - \text{Re}^{\frac{3}{2}} \frac{h^3}{12} \frac{\partial^4 v}{\partial y^4} \quad \text{as } h \rightarrow 0, \quad (3.21)$$

and for the  $x$ - and  $y$ -momentum equations:

$$\tau_h = \frac{h^2}{3} u \frac{\partial^3 u}{\partial x^3} - \text{Re}^{\frac{3}{2}} \frac{h^3}{12} v \frac{\partial^4 u}{\partial y^4} + \frac{h^3}{12} \frac{\partial^4 \varphi}{\partial x^4} + \frac{h^2}{12} \left( \text{Re}^{-1} \frac{\partial^4 u}{\partial x^4} + \text{Re} \frac{\partial^4 u}{\partial y^4} \right), \quad (3.22)$$

and

$$\tau_h = \frac{h^2}{3} u \frac{\partial^3 v}{\partial x^3} - \text{Re}^{\frac{3}{2}} \frac{h^3}{12} v \frac{\partial^4 v}{\partial y^4} + \text{Re}^{\frac{5}{2}} \frac{h^3}{12} \frac{\partial^4 \varphi}{\partial y^4} + \frac{h^2}{12} \left( \text{Re}^{-1} \frac{\partial^4 v}{\partial x^4} + \text{Re} \frac{\partial^4 v}{\partial y^4} \right), \quad (3.23)$$

respectively, as  $h \rightarrow 0$ . Choosing the mesh width  $h$  in the boundary layer such that  $\text{Re}^{\frac{1}{2}} h = \mathcal{O}(1)$ , helps to control the growth of the truncation error of the continuity equation and the  $x$ -momentum equation. This can be seen as follows, choosing

$$h := \frac{2\sigma}{N} \quad \text{and} \quad \sigma := \frac{C}{\sqrt{\text{Re}}}, \quad (3.24)$$

yields

$$\text{Re}^{\frac{1}{2}} h = \text{Re}^{\frac{1}{2}} \frac{2}{N} \frac{C}{\text{Re}^{\frac{1}{2}}} = \frac{2C}{N}, \quad (3.25)$$

which is independent of the Reynolds number. Note, that by setting  $C := N/2$ , instead of an arbitrarily chosen constant  $C$ ,  $\text{Re}^{\frac{1}{2}} h$  exactly equals one. (This choice

implies a genuine Shishkin mesh,  $\sigma = \sigma(\varepsilon, N)$ , instead of a piecewise-uniform mesh,  $\sigma = \sigma(\varepsilon)$ .)

However, both choices ( $C$  is constant and  $C = N/2$ ) do not bound the truncation error of the  $y$ -momentum equation, for increasing  $\text{Re}$  with  $h$  fixed. The coefficient of the  $\frac{\partial^4 \varphi}{\partial y^4}$ -term in (3.23) can be written as  $\text{Re}(\text{Re}^{\frac{1}{2}} h)^3$ , which shows a linear increase with increasing Reynolds number as long as (3.24) holds. A further expansion of the Taylor-series for the discrete  $\frac{\partial \varphi}{\partial y}$ -term reveals that each term in the expansion contains the coefficient  $\text{Re}(\text{Re}^{\frac{1}{2}} h)^p$ , with  $p$  the order of the truncation term.

In order to control this term, the mesh should be chosen such that

$$\text{Re}^{\frac{5}{2}} h^3 = \mathcal{O}(1). \quad (3.26)$$

This can be achieved by choosing

$$h := \frac{2\sigma}{N} \quad \text{and} \quad \sigma := \frac{C}{\text{Re}^\alpha}, \quad \alpha \in \mathbb{Q}^+, \quad (3.27)$$

where  $C$  is, again, a suitably chosen constant. From this, one can derive that for  $\alpha = \frac{5}{6}$ , (3.26) is satisfied. E.g.,

$$\sigma := \frac{C}{\text{Re}^{\frac{5}{6}}} \quad (3.28)$$

might be a well-suited alternative for (3.11). Again, if  $C := \frac{N}{2}$  then (3.26) is equal to 1.

The convergence history for the residuals is still shown in Figure 3.10. As in the case of the mesh width variation, with a fixed Reynolds number, only the  $R_x$ ,  $R_y$  and the the  $R_{\text{div}u}$  components of the residual vector are important. This figure shows that the efficiency of the Navier-Stokes method is not parameter uniform, i.e., the computational work depends on the Reynolds number.

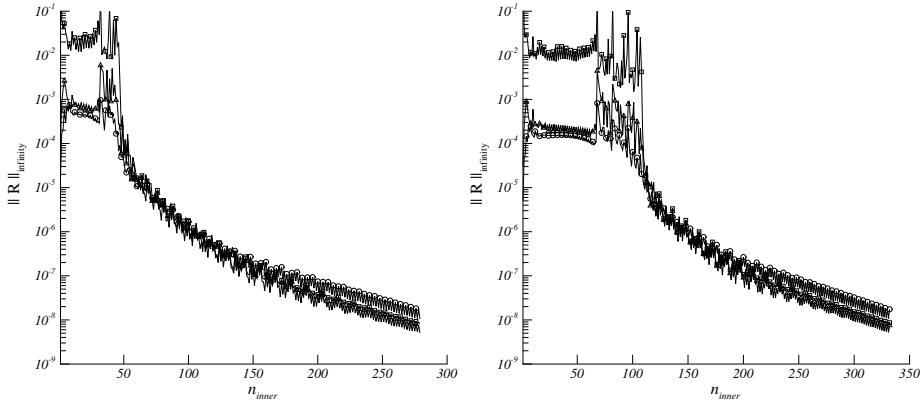


Figure 3.10: Convergence histories of the iteration process on  $\Omega^{P,4N}$  for  $\text{Re} = 4.10^6$  (left figure) and  $\text{Re} = 16.10^6$  (right figure).  $\mathbf{R}$  is the residual of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ) and the  $y$ -momentum equation ( $\circ$ ), only every fourth marker is shown. ( $N = 20$ ).

**Triple-layered piecewise-uniform mesh** From Figure 3.9 can be observed that the  $v$ -component of the Blasius solution has two regions where rather steep gradients are present, i.e., a lower layer which, for  $\text{Re} = 10^6$ , is located in, say,  $0 \leq \eta \leq 3$  and an upper layer located in, say,  $3 \leq \eta \leq 6$ . In the construction of  $\Omega^{P,4N}$ , which is based upon the boundary layer in the  $u$ -component, this lower layer has been accounted for, collectively with the upper layer, not separately. We will now proceed by resolving the lower layer more accurately than the upper layer. To do so we propose the triple-deck piecewise-uniform mesh depicted in Figure 3.11. We denote this mesh by  $\Omega^{TD,N}$ .

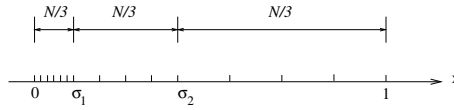


Figure 3.11: Example of a triple-deck piecewise-uniform mesh.

This new grid is defined by *two* transition parameters,  $\sigma_1$  and  $\sigma_2$ , which are specified by

$$\sigma_1 = \frac{C_1}{\text{Re}^{\frac{5}{6}}}, \quad C_1 = 200, \quad \text{and} \quad \sigma_2 = \frac{C_2}{\text{Re}^{\frac{1}{2}}}, \quad C_2 = 25. \quad (3.29)$$

Given the analytical results from the previous section, we speculate that for our reduced Navier-Stokes equations, the lower-layer thickness is proportional to  $\text{Re}^{-\frac{5}{6}}$ . The constant  $C_1$ , in the expression for  $\sigma_1$ , is determined by the upper bound chosen for the lower layer. We choose  $\eta = 2\sqrt{2}$  for  $\text{Re} = 10^6$ . Now each region is discretized uniformly using  $\frac{N}{3}$  grid points. In this section we repeat the computations of the previous section but now performed on  $\Omega^{TD,N}$  ( $N = 90$ ). The computed (scaled) velocity components are shown in Figure 3.12.

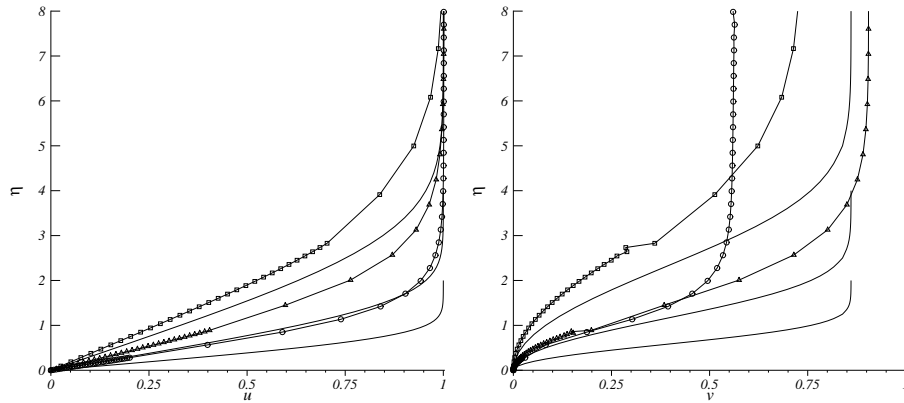


Figure 3.12:  $u$ -profile (*left figure*) and  $v$ -profile (*right figure*) at center of plate, computed on  $\Omega^{TD,N}$ , for  $\text{Re} = 10^6$  ( $\square$ ),  $\text{Re} = 4 \times 10^6$  ( $\triangle$ ) and  $\text{Re} = 16 \times 10^6$  ( $\circ$ ), and the Blasius solution (solid). The vertical scale corresponds with  $\text{Re} = 10^6$ . ( $N = 90$ ).

The numerical results reveal a more or less similar dependence on the Reynolds number, as can be concluded from

$$\frac{\delta_4}{\delta_1} = 0.63, \quad \text{and} \quad \frac{\delta_{16}}{\delta_4} = 0.62. \quad (3.30)$$

However, the sensitive  $v$ -component suggests that the triple-deck grid has a negative influence on the solution quality; the  $v$ -component shows a non-monotonous convergence behaviour. The accuracy gained by the increased resolution of the lower layer is more than undone by the jump in the mesh size at  $y = \sigma_1$  (which is located inside the boundary layer instead of outside like the jump at  $y = \sigma_2$ ), and also by the somewhat coarser grid in the upper layer as compared to the double-deck grid. In the remainder of this chapter we refrain from application of the triple-deck piecewise-uniform mesh.

The convergence histories of the residuals, for the computations on  $\Omega^{TD,N}$ , are still shown in Figure 3.13.

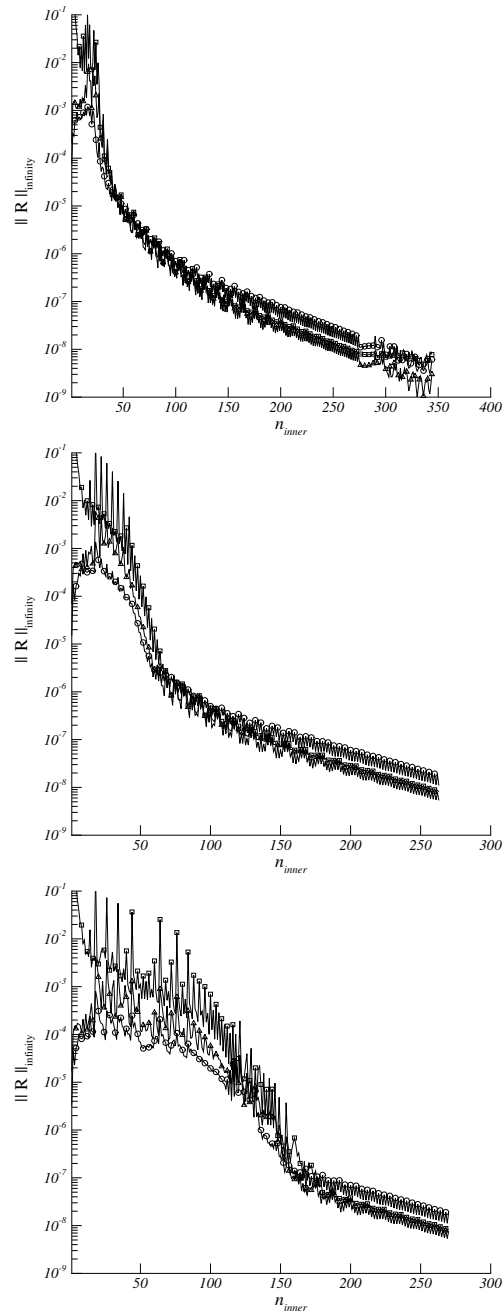


Figure 3.13: Convergence histories of the iteration process on  $\Omega^{TD,N}$  for  $\text{Re} = 10^6$  (top graph),  $\text{Re} = 4 \times 10^6$  (middle graph) and  $\text{Re} = 16 \times 10^6$  (bottom graph).  $\mathbf{R}$  is the residual of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ) and the  $y$ -momentum equation ( $\circ$ ), only every fourth marker is shown. ( $N = 90$ ).

### 3.3.3 Flat plate with free-surface perturbation

As mentioned in Section 3.1, this test case serves to model a typical flow problem arising from numerical ship hydrodynamics, however, without the geometrical complexity. Here the free-surface flow is added to the flow problem of the previous (two) sections. The free-surface waves are generated by (i) a perturbation imposed at the free surface and (ii) the displacement effect of the boundary layer. We conjecture that the latter is negligible compared to the effects of the imposed pressure perturbation. The computed wave pattern as a result of the displacement effect only, can be found in [10]. The perturbation imposed is again of the form (2.21). The solution and convergence behaviour of the free-surface flow problem, without the presence of the flat plate, have already been studied in the previous chapter. Here the flat plate, with length  $\ell = 1$ , introduces a boundary layer and wake which now interacts with the free surface. This flow problem, unlike the flow problem from the previous sections, has no known analytical solution. Since the influence of the boundary-layer solution is only local, it can be assumed that the flow solution outside the boundary layer corresponds to the solution computed in the previous chapter, at least qualitatively.

This brings us to the item of matching the boundary layer with the free surface. Most numerical free-surface methods suffer from the ‘contact-line’ problem as discussed in Section 3.1. In order to circumvent the contact-line problem, any free-surface update procedure should preferably *not* involve the velocity. Since our free-surface update procedure is based on the pressure, in principle, it does not suffer from this drawback. This can be seen as follows, substitution of the ‘no-slip’ condition into (2.22) yields no incompatibility as long as  $\|\nabla\varphi\|$  remains finite.

Here we consider as value for the amplitude of the perturbation:  $P = 0.05$ , with  $\alpha = -4$ ,  $(x_c, y_c) = (-\frac{1}{2}, 0)$ . The Reynolds and Froude number for this computation are set at  $\text{Re} = 10^6$  and  $\text{Fr} = 0.6$ , respectively. And the boundary layer is considered to be turbulent with an eddy-viscosity  $\nu_t$  determined by an algebraic turbulence model. Here the Cebeci-Smith model is used. Details of this model and its implementation can be found in [31]. The flow computations are performed on  $\Omega^{E,4N}$  and  $\Omega^{P,4N}$ . Both domains are of the size  $(4\ell)^3$ , with  $\ell$  being the plate length. All computations are performed with an  $\mathcal{O}(h^2)$ -upwind biased difference scheme for the quasi free-surface condition and the nonlinearity of the quasi free-surface condition is taken care of via Newton’s method. As mentioned in Section 3.1, the free-surface algorithm corresponds to the one described in Section 2.3.1.

The result of the interaction of the boundary layer with the free-surface flow can be seen in Figure 3.14 for the computation on  $\Omega^{E,4N}$ .

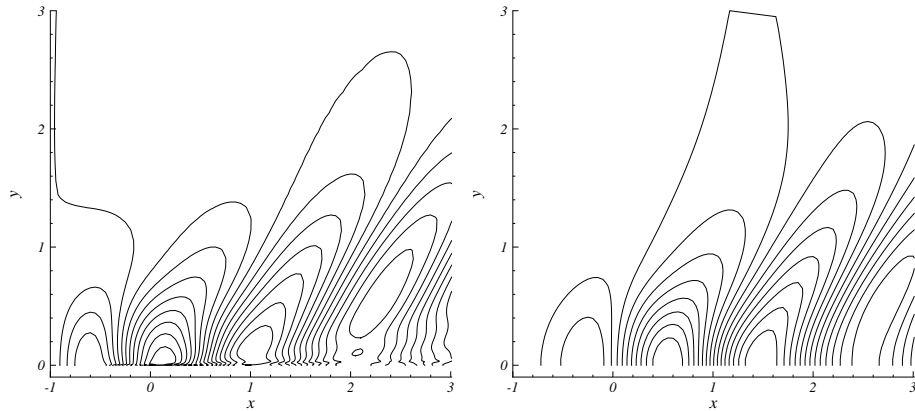


Figure 3.14: Computed wave elevation on  $\Omega^{E,4N}$  for  $P = 0.05$  with flat plate (*left figure*) and without flat plate (*right figure*). ( $N = 20$ ). Note that in the *left figure*  $(x_c, y_c) = (-\frac{1}{2}, 0)$  and in the *right figure*  $(x_c, y_c) = (0, 0)$ .

The presence of the boundary layer and wake, due to the flat plate, is clearly visible in the region near the plane of symmetry,  $y = 0$ . As anticipated, the solution away from the flat plate does not seem to be influenced very much by the presence of the flat plate. A similar conclusion can also be drawn from observation of the wave pattern computed on  $\Omega^{P,4N}$ . A comparison of the solutions, computed on  $\Omega^{E,4N}$  and  $\Omega^{P,4N}$ , reveals a difference of the free-surface flow, as can be seen in Figure 3.15.

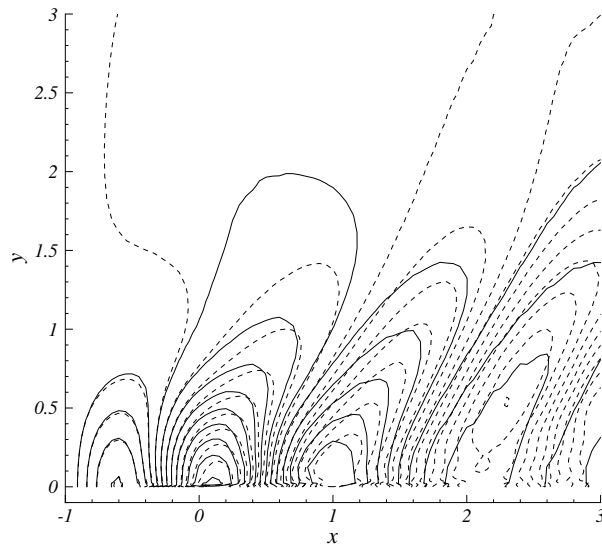


Figure 3.15: Computed wave elevation on  $\Omega^{E,4N}$  (*dashed*) and on  $\Omega^{P,4N}$  (*solid*). ( $N = 20$ ).



Figure 3.15 shows a phase difference between the two solutions in the outer-flow region. This phase difference can be attributed to the sudden increase of the mesh width on  $\Omega^{P,4N}$ . This behaviour is consistent with the numerical results presented in Section 2.4.4. There it was shown that a dispersion behaviour changes when the mesh width is varied. Furthermore, the sudden change in mesh width also increases the numerical damping, resulting in a loss of amplitude for the computed wave pattern as compared to the solution computed on  $\Omega^{E,4N}$  as can be seen in Figure 3.16. Note the large difference in the wave amplitude in the downstream region of the flat plate.

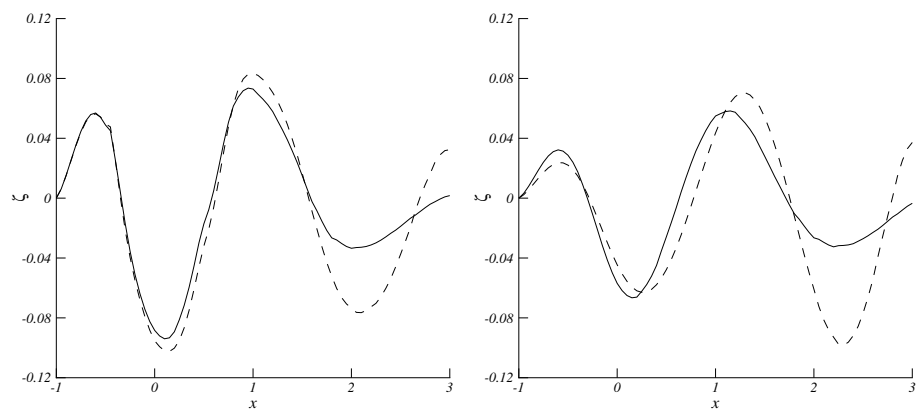


Figure 3.16: Computed wave elevation on  $\Omega^{E,4N}$  (*dashed*) and on  $\Omega^{P,4N}$  (*solid*) for  $y = 0$  (*left figure*) and  $y = 0.5$  (*right figure*). ( $N = 20$ ).

The two types of grids also have an effect on the computed velocity profiles in the boundary layer. The (unscaled)  $u$ - and  $v$ -velocity profiles are shown in Figure 3.17. Note that the  $v$ -profile shows a non-monotone behaviour at  $y = \sigma$ , but not the  $u$ -profile.

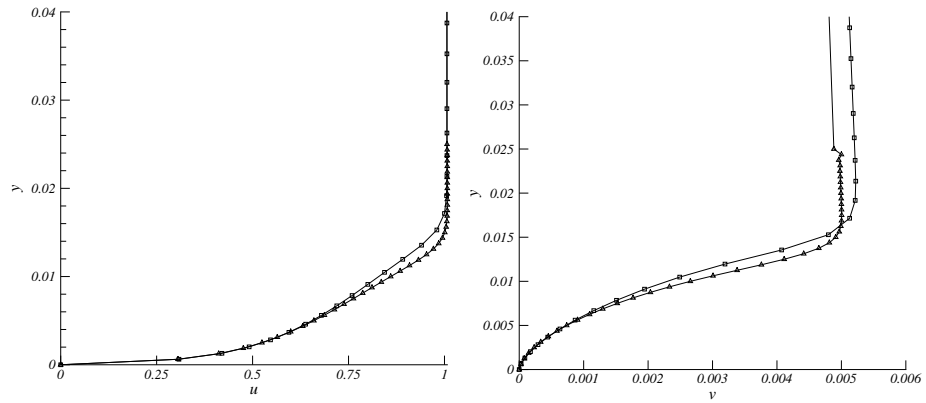


Figure 3.17: Computed (unscaled)  $u$ -profile (left figure) and  $v$ -profile (right figure) in the boundary layer at the center of the plate.  $\Omega^{P,4N}$  ( $\Delta$ ) and  $\Omega^{E,4N}$  ( $\square$ ). ( $N = 20$ ).

Since  $\Omega^{P,4N}$  has more grid points near the outer edge of the boundary layer, on  $\Omega^{P,4N}$  the curvature in the  $u$ -profile is better captured than on  $\Omega^{E,4N}$ .

The convergence behaviour for the computations on  $\Omega^{P,4N}$  and  $\Omega^{E,4N}$  is shown in Figures 3.18 and 3.19, respectively.

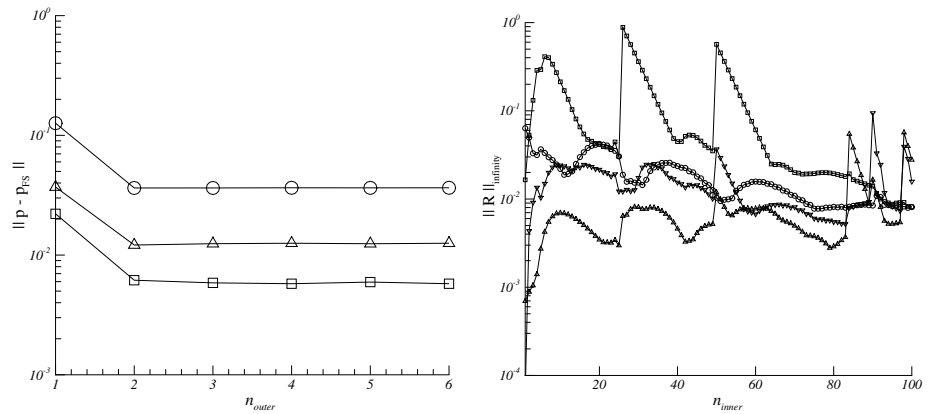


Figure 3.18: Convergence histories on  $\Omega^{P,4N}$ . *Left:* of the free-surface pressure defect; measured in  $L_1$ -norm ( $\square$ ),  $L_2$ -norm ( $\Delta$ ), and the  $L_\infty$ -norm ( $\circ$ ). *Right:* of the inner-iteration process,  $\mathbf{R}$  is the residual of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ), the  $y$ -momentum equation ( $\circ$ ), and the  $z$ -momentum equation ( $\nabla$ ). ( $N = 20$ ).

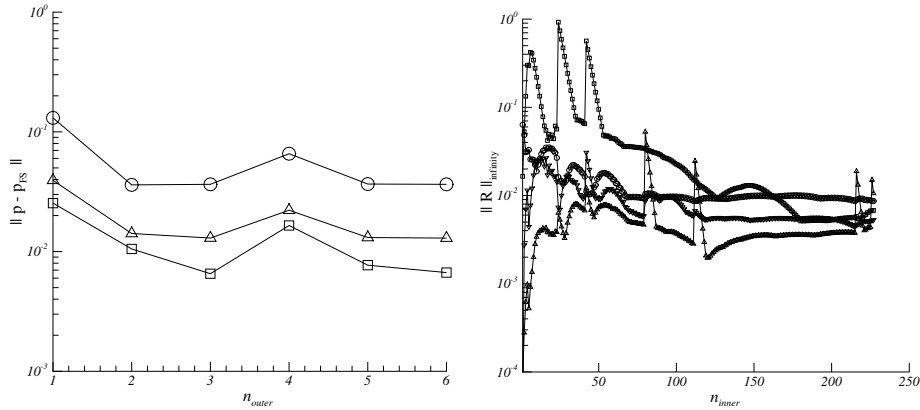


Figure 3.19: Convergence histories on  $\Omega^{E,4N}$ . *Left*: of the free-surface pressure defect; measured in  $L_1$ -norm ( $\square$ ),  $L_2$ -norm ( $\Delta$ ), and the  $L_\infty$ -norm ( $\circ$ ). *Right*: of the inner-iteration process,  $\mathbf{R}$  is the residual of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ), the  $y$ -momentum equation ( $\circ$ ), and the  $z$ -momentum equation ( $\nabla$ ). ( $N = 20$ ).

In these computations the grid-sequencing strategy has been applied. This strategy is responsible for the jumps in  $\|\mathbf{R}\|_\infty$  at  $n_{inner} = 5, 25$  and  $50$  for the computation on  $\Omega^{P,4N}$  and at  $n_{inner} = 5, 24$  and  $42$  for the computation on  $\Omega^{E,4N}$ . The grid updates occur at  $n_{inner} = 83, 89, 91, 92$  and  $95$  for  $\Omega^{P,4N}$  and at  $n_{inner} = 79, 111, 112, 215, 225$  and  $227$  for  $\Omega^{E,4N}$ . As can be observed from both figures, only the first free-surface update results in a substantial decrease of the pressure defect, the following updates do practically nothing. This can be explained by the fact that these latter free-surface updates induce perturbations which are of the order of accuracy with which the stationary Navier-Stokes subproblems are being solved. Furthermore, note the difference in computational work for the two types of meshes. The computation on  $\Omega^{P,4N}$  requires again less work than that on  $\Omega^{E,4N}$ .

### Mesh adaptation strategy

The main advantage of the piecewise-uniform mesh is its simplicity. This type of mesh lends itself perfectly well to be incorporated into an a-posteriori mesh adaptation method. The key issue in the construction of the piecewise-uniform mesh is the determination of  $\sigma$ , i.e., the location of the transition between the coarse submesh and the fine submesh. In this test case, and for the flow computation in numerical ship hydrodynamics in general, the boundary-layer thickness will vary in streamwise and also in depth direction, say the  $x$ - and  $z$ -direction. Hence, to accurately capture the boundary layer,  $\sigma := \sigma(x, z)$ . Furthermore, the value of  $\sigma$  is determined by some discrete measure of the boundary-layer thickness, e.g.,

$$\sup_{y \geq \sigma} u(x, y, z) \leq \gamma, \quad \text{with } \gamma \approx 1, \quad (3.31)$$

with  $u$  scaled. Once  $\sigma$  has been determined one has to choose the fractions of grid points to be distributed over the coarse and fine submesh. (Here each submesh contained  $\frac{N}{2}$  mesh points.) Then the piecewise-uniform mesh is completely specified.

### 3.4 Conclusions

The subject of this chapter was the investigation of the suitability of piecewise-uniform meshes for Navier-Stokes computations with free-surface water waves.

The presence of a boundary layer requires the necessity of a non-uniform mesh. Here we considered two variants, i.e., the piecewise-uniform mesh, a simplified form of the Shishkin-mesh, and the standard exponentially-stretched mesh. On both types of meshes we first computed the laminar boundary layer, neglecting the free-surface flow. Variations of the mesh width and Reynolds number reveal a comparable behaviour of the solution computed on both types of meshes. However, on the piecewise-uniform mesh a non-smooth behaviour of the pressure and the normal velocity component occurs at the interface between the fine and the coarse mesh. (The streamwise velocity component does not show this non-smooth behaviour.) This erroneous behaviour of the pressure and the normal velocity component does not occur in the solution computed on the exponentially stretched mesh.

Error analysis for the discrete Navier-Stokes equations according to the finite difference schemes described in Section 2.3.2 shows that for increasing Reynolds number and fixed mesh size, in order to bound the local truncation error in the  $y$ -momentum equation, it makes sense to take  $\sigma \div \text{Re}^{-\frac{5}{8}}$  instead of  $\sigma \div \text{Re}^{-\frac{1}{2}}$ . Numerical computations with a triple-layered, piecewise uniform mesh have not (yet) confirmed this.

As mentioned previously, an advantage of the piecewise-uniform mesh – which in fact motivates this research – is that its generation is so simple, and therefore also its a-posteriori adaptation to the turbulent boundary layer, an adaptation which may be easily done simultaneously with that of the free surface. For the present flat-plate case, adaptation of the mesh to the boundary layer appeared to be not necessary yet.

Given the aforementioned disadvantage of a non-smooth solution behaviour near the jump in mesh width, the piecewise-uniform mesh will not be considered anymore in this thesis.

## CHAPTER 4

# Numerical Simulation of 3D Steady Free-Surface Flows around a Ship Hull

### 4.1 Introduction

The drag of large merchant and naval vessels is of great economical, environmental and hence societal importance. The main contributions to the drag of the underwater part of the hull are: *viscous* drag associated with the boundary layer and wake, and *wave* drag due to the excitation of a wave pattern. While at relatively low speeds the viscous drag is dominant. The wave drag becomes increasingly important at higher speeds and can amount up to 50 % of the total drag in still water.

An accurate estimate of the drag requires an accurate description of the complete flow field, i.e., the shape of the free surface and the underlying velocity field and pressure. Traditionally, the computation of the complete flow field is performed through a segregated approach justified by the validity of Froude's hypothesis; i.e., gravity and viscosity are supposed to act independently for smooth, slender hull shapes and their mutual interaction is supposed to be a higher-order effect. This decomposition entailed inviscid boundary-integral methods for the computation of the free surface, see, e.g., [60] and the references therein, and boundary-layer/Navier-Stokes methods for the computation of the viscous effects, see, e.g., [31]. This splitting of the two effects is justified as long as the viscous effects do not strongly interact with the free-surface flow. The approach fails if, e.g., massive separation of the boundary layer occurs near or at the free surface. To properly, i.e. simultaneously, account for the influences of both features the (Reynolds-Averaged) Navier-Stokes equations have to be solved in combination with the free-surface flow conditions. This is a major subject within the field of numerical ship hydrodynamics for a number of years now. Much progress has already been made into this direction, see, e.g., [61] for a short overview.

The main topic of this chapter is the application of the free-surface algorithm, as described in the previous chapters, to a practical application originating from numerical ship hydrodynamics. The test case to be considered comprises the computation of the complete (turbulent) flow field, including the generated wave pattern, around

the Series 60 hull (Figure 4.1) at  $Fr = 0.316$  and  $Re = 10^6$ . This is a standard test case in numerical ship hydrodynamics for free-surface flow algorithms. Much is known about the flow solution. In this chapter we will concentrate on the free-surface flow and on the interaction of the free surface with the boundary layer and wake.

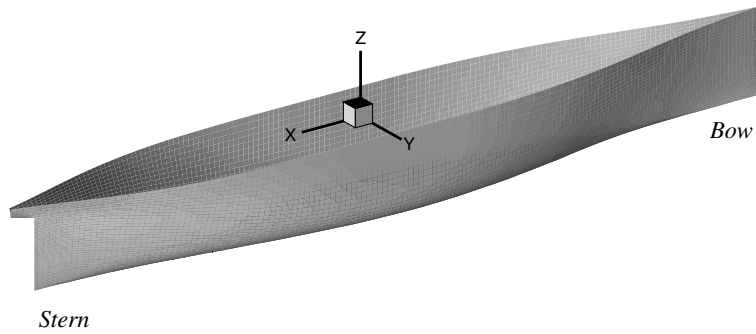


Figure 4.1: The Series 60 hull.

The outline of this chapter is as follows. Section 4.2 addresses the computational method, which is an extension to curvilinear domains of the Cartesian-domain method used in Chapter 2. Of particular interest in Section 4.2 is the remeshing algorithm. This is a non-trivial matter in 3D and, particularly, in curvilinear domains because of the danger of, e.g., mesh tangling. Section 4.3 contains the numerical results, applying a 'double-body' approximation, a partially linearized formulation as well as a fully nonlinear one. Section 4.4 concludes this chapter.

## 4.2 Computational method

### 4.2.1 Discretization in curvilinear domain

Many flow problems arising from engineering applications involve a complex geometry and as a result the discretization of the computational domain and corresponding equations is far from trivial. The discretization of the computational domain entails a distribution of a finite set of points, in case of a finite-difference method. Here we apply a structured boundary-fitted mesh, instead of a Cartesian or an unstructured mesh, in order to achieve *(i)* proper resolution of the steep gradients near the hull, *(ii)* easy implementation of the boundary conditions and *(iii)* convenience in programming. However, for cases involving a very complex geometry, generation of a structured boundary-fitted mesh can be very time consuming. An elaborate discussion on the advantages and disadvantages of the different types of meshes can be found in [79] and the references therein.

Here the generation of the structured boundary-fitted mesh is obtained via a

non-singular mapping

$$\mathcal{T} : \mathbf{x} = \mathbf{x}(\boldsymbol{\xi}), \quad (4.1)$$

where  $\mathbf{x}$  denotes the Cartesian physical coordinates and  $\boldsymbol{\xi}$  the curvilinear computational coordinates, and where  $\mathcal{T}$  maps the physical domain  $\Omega$  onto the rectangular hexahedron  $\mathcal{G}$ . The Jacobian of the mapping is required to be finite:

$$\sqrt{g} := \left| \frac{\partial x^i}{\partial \xi^j} \right| < \infty, \quad i, j = 1, 2, 3. \quad (4.2)$$

Once the mapping has been established, the discretization of  $\mathcal{G}$ , here denoted as  $\mathcal{G}_h$ , is obtained by defining the following mesh points located at  $\boldsymbol{\xi}_j$ ,  $\mathbf{j} = (j_1, j_2, j_3)^T$ :

$$\mathcal{G}_h = \{ \boldsymbol{\xi} : \xi_{j_\alpha}^\alpha = j_\alpha \Delta \xi^\alpha, j_\alpha = 1, \dots, N_\alpha, \alpha = 1, 2, 3 \}, \quad (4.3)$$

with  $N_\alpha \in \mathbb{N}$  denoting the number of grid points in the  $\xi^\alpha$ -direction. Here the  $\xi^1$ -direction corresponds to the main-flow direction, the  $\xi^2$ -direction corresponds to a direction normal to the hull and the  $\xi^3$ -direction is tangential to the hull. The details of the grid layout are discussed in Chapter 3 of [31].

#### 4.2.2 Governing equations in curvilinear coordinates and discretization

The physical laws, which govern the free-surface flow problem, are tensor laws and, hence, invariant under coordinate transformation. The complete flow problem in this chapter is governed by the equations given in Section 2.2. The discretization of the flow problem requires a decomposition of the vector equation into its components. This is, in the case of a curvilinear coordinate system, a non-trivial matter. In the remainder of this chapter the tensor notation is applied. The uninitiated reader is referred to [7] for an elaborate introduction to the subject. The equations for conservation of mass and momentum, written in curvilinear coordinates and for a spatially invariant eddy-viscosity assumption for the Reynolds stresses, read

$$\frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^j} (U^j) = 0, \quad (4.4)$$

and

$$\frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^j} (\sqrt{g} T^{jk} \mathbf{a}_j) = 0. \quad (4.5)$$

Here  $T^{jk}$  represents the momentum flux vector and is given by

$$T^{jk} := U^j U^k + g^{jk} \varphi - \nu \left( g^{jm} U_{,m}^i + g^{ik} U_{,k}^j \right). \quad (4.6)$$

$U^j$  denotes the contravariant velocity component,  $\mathbf{a}_i$  the covariant base vector and  $g^{jk} = \mathbf{a}^j \cdot \mathbf{a}^k$  is the contravariant version of the metric tensor.

The discrete system of equations is obtained by replacing the differential operators by finite differences. The order of accuracy of the various parts of the system of equations, i.e., convection, diffusion and the pressure gradient, is the

same as described in Section 2.3.2. This yields the following expressions for the discrete differential operators: the  $\xi^1$ -derivative, in the continuity equation and in the convection term in the momentum equations, is replaced by

$$D_{\xi^1}^- = \frac{3}{2}E_{\xi^1}^0 - 2E_{\xi^1}^{-1} + \frac{1}{2}E_{\xi^1}^{-2}, \quad (4.7a)$$

and the  $\xi^2$ - and  $\xi^3$ -derivatives by

$$D_{\xi^\alpha}^- = \frac{1}{3}E_{\xi^\alpha}^{+1} + \frac{1}{2}E_{\xi^\alpha}^0 - E_{\xi^\alpha}^{-1} + \frac{1}{6}E_{\xi^\alpha}^{-2}, \quad \alpha = \{2, 3\}, \quad (4.7b)$$

where  $E_{\xi^\alpha}^k u \equiv u_{j_\alpha+k}^\alpha$  is the linear shift operator. For the pressure gradient in the momentum equations an opposite bias is applied, i.e.,

$$D_{\xi^\alpha}^+ = -\frac{1}{3}E_{\xi^\alpha}^{-1} - \frac{1}{2}E_{\xi^\alpha}^0 + E_{\xi^\alpha}^{+1} - \frac{1}{6}E_{\xi^\alpha}^{+2}, \quad \alpha = \{1, 2, 3\}. \quad (4.7c)$$

The discrete viscous operator only consists of the  $\xi^2$ - and  $\xi^3$ - derivatives and is discretized by second-order central difference schemes, see pages 64 – 65 in [31] for a clear description. The geometric quantities are approximated by central differences. For example, the  $\mathbf{a}_1$  covariant base vector is in discrete form approximated by

$$(\mathbf{a}_1)_{i,j,k} = \begin{pmatrix} \frac{1}{2}(x_{i+1,j,k}^1 - x_{i-1,j,k}^1) \\ \frac{1}{2}(x_{i+1,j,k}^2 - x_{i-1,j,k}^2) \\ \frac{1}{2}(x_{i+1,j,k}^3 - x_{i-1,j,k}^3) \end{pmatrix}. \quad (4.8)$$

Once the discrete covariant base vectors are available, the discrete contravariant base vector can be computed from

$$(\mathbf{a}^l)_{i,j,k} = (\mathbf{a}_m)_{i,j,k} \times (\mathbf{a}_n)_{i,j,k}, \quad (4.9)$$

with  $l, m, n$  in cyclic order. The discrete Jacobian  $\sqrt{g}$  follows from

$$(\sqrt{g})_{i,j,k} = (\mathbf{a}_l \cdot (\mathbf{a}_m \times \mathbf{a}_n))_{i,j,k}, \quad (4.10)$$

again with  $l, m, n$  in cyclic order. After the discretization a change of variables is applied, i.e., the contravariant velocity components are replaced by the grid-aligned physical velocity components  $U(m)$  according to

$$U(m) = U^m |\mathbf{a}_m| \quad (\text{no sum over } m). \quad (4.11)$$

The modified system for  $(U(1), U(2), U(3), \varphi)^T$  yields better solution properties (better condition) than the original discrete system.

The transformation of the quasi free-surface condition to curvilinear coordinates can be derived as follows. In vector notation the quasi free-surface condition reads

$$\mathbf{u} \cdot \nabla \varphi - \text{Fr}^{-2} \mathbf{u} \cdot \mathbf{e}_z = 0, \quad (4.12)$$



where we have assumed that the atmospheric pressure  $p = 0$  holds at the free surface. Application of the contravariant velocity components and the transformation of the gradient of the pressure, i.e.,

$$\frac{\partial \varphi}{\partial x^\alpha} = a_{(\alpha)}^\beta \frac{\partial \varphi}{\partial \xi^\beta}, \quad (4.13)$$

yield

$$U^\alpha \frac{\partial \varphi}{\partial \xi^\alpha} - \text{Fr}^{-2} U^\alpha a_\alpha^{(3)} = 0. \quad (4.14)$$

Here we have used the fact that  $\mathbf{a}_\beta \cdot \mathbf{a}^\gamma = \delta_\beta^\gamma$ . The last step in the discretization process is to convert to the grid-aligned physical velocity components, as defined in (4.11). The quasi free-surface condition written in these variables reads

$$\frac{U(\alpha)}{|\mathbf{a}_\alpha|} \frac{\partial \varphi}{\partial \xi^\alpha} - \text{Fr}^{-2} U^\alpha a_\alpha^{(3)} = 0 \quad (\text{sum over } \alpha). \quad (4.15)$$

The nonlinearity of the quasi free-surface condition is dealt with via the SOR-Newton method. Throughout this chapter, for convergence reasons (to be analyzed in Chapter 5), the pressure gradient in the quasi free-surface condition has to be discretized with the  $\mathcal{O}(h)$  upwind scheme, i.e.,  $\frac{\partial}{\partial \xi^\alpha}$  is replaced by  $D_{\xi^\alpha}^-$  corresponding to  $E_{\xi^\alpha}^0 - E_{\xi^\alpha}^{-1}$  for  $\alpha = \{1, 2, 3\}$ , respectively.

### 4.2.3 Remeshing algorithm

In the free-surface algorithm, as described in the Chapter 2, we solve (either fully or partially) a series of steady Navier-Stokes boundary-value problems with the quasi free-surface condition imposed on the (approximate) location of the free surface. Each iterate of the steady Navier-Stokes boundary value problem, i.e., either fully converged, corresponding to the algorithm in Section 2.3.1, or after a single relaxation sweep only, corresponding to the algorithm in Section 2.5.1, generates a new location of the free surface. In general this new free-surface location necessitates a smoothing operation on all other mesh points. In this thesis this smoothing is performed using a method based upon the concept of *moving-mesh* methods, also known as *r-refinement* (the *r* referring to *redistribution* or *relocation*), see e.g., [72, 82]. Since we are considering stationary problems only, the grid velocities are zero. One of the disadvantages of a moving-mesh approach is the possible occurrence of ‘mesh tangling’ (Figure 4.2).

The algorithm which is employed to redistribute the underlying mesh points is based on the principle of *equidistribution*, see e.g., [62]. This method was originally derived for time-dependent PDE’s, where the discrete equidistribution principle yields the location of the mesh points according to

$$\Delta x_i(t) \mathcal{M}_i(x_i, t) = \text{constant}, \quad (4.16)$$

where  $\Delta x_i = x_{i+1} - x_i$  and where  $\mathcal{M}_i(x_i, t)$  is a suitably chosen weight function, usually depending on the first- and second-order derivatives of the PDE solution.

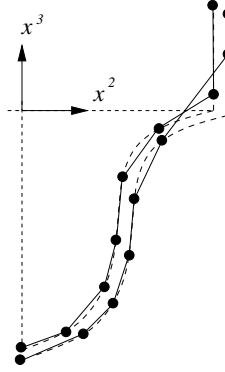


Figure 4.2: Example of crossing grid lines belonging to different families (mesh tangling). (The dashed lines denote the initial grid.)

The redistribution algorithm employed in this thesis is described below. Let  $\xi_0$  be the parameterization of a curve, corresponding to the  $\xi^3$ -direction in  $\mathcal{G}_h$ , through the set of discrete grid points  $\xi_0(j_3)$ ,  $j_3 = 1, \dots, N_3$  belonging to the initial grid in the physical space, see Figure 4.3.

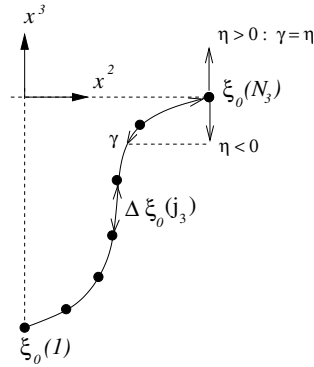


Figure 4.3: Definition of  $\gamma$ ,  $\xi_0$  and  $\Delta\xi_0$  in the physical space. For  $\eta > 0$  :  $\gamma = \eta$ . For  $\eta < 0$  :  $\gamma$  is the length of the corresponding piece of the arc.

Let  $\xi_0(1) = 0$  and  $\xi_0(N_3) = \ell_0$  indicate the beginning and the end of the curve, where  $\ell_0$  is the arclength of the curve:

$$\ell_0 = \sum_{j_3=1}^{N_3-1} \Delta\xi_0(j_3), \quad (4.17)$$

and where

$$\Delta\xi_0(j_3) := \xi_0(j_3 + 1) - \xi_0(j_3), \quad j_3 = 1, \dots, N_3 - 1 \quad (4.18)$$

denotes the distance, along the curve, between two subsequent grid points. Given a new location of the free surface in the physical space, here denoted by  $\eta$ , the computation of  $\xi_n(j_3)$ ,  $j_3 = 1, \dots, N_3 - 1$  follows from (4.16) with  $\mathcal{M} = 1$ . The distribution of the discrete grid points along the curve for the adapted grid follows from

$$\begin{aligned} \frac{\Delta\xi_n(j_3)}{\ell_0 + \gamma} &= \frac{\Delta\xi_0(j_3)}{\ell_0} \Leftrightarrow \\ \Delta\xi_n(j_3) &= \left(1 + \frac{\gamma}{\ell_0}\right) \Delta\xi_0(j_3), \end{aligned} \quad (4.20)$$

where  $\gamma$  is defined as in Figure 4.3 and the displacement of each grid point with respect to its initial position can be computed from

$$\xi_n(j_3) - \xi_0(j_3) = \frac{\gamma}{\ell_0} \sum_{i=1}^{j_3-1} \Delta\xi_0(i). \quad (4.21)$$

The final step is to determine the  $(x^2, x^3)$  coordinates for the given displacement (4.21) of the grid point along the curve. Here we assume that the grid points are connected by piecewise linear segments. Hence, the new coordinates can be found from

$$\frac{x_n^\alpha(j_3) - x_0^\alpha(j_3)}{x_0^\alpha(j_3 + 1) - x_0^\alpha(j_3)} = \frac{\xi_n(j_3) - \xi_0(j_3)}{\Delta\xi_0(j_3)}, \quad j_3 = 1, \dots, N_3 - 1, \quad \alpha = \{1, 2\}, \quad (4.22)$$

where  $x_n^\alpha, x_0^\alpha$  represent the coordinates of the grid points in the new grid and the initial grid, respectively.

### 4.3 Numerical results

This section contains the numerical results as obtained for the Series 60 hull. The Froude and Reynolds number, for all computations in this chapter, are set at 0.316 and  $10^6$ , respectively. The first subsection will present the results for the case of a so-called ‘double-body’ approximation, as well as for a ‘uniform-flow linearization’. In both approximations it is assumed that at the free surface the perturbations due to the hull are small. No free-surface updates are computed in this first subsection. In the second subsection the fully nonlinear free-surface flow is addressed, as well as free-surface updates.

The computational domain, denoted by  $\Omega_h$ , extends from  $\frac{\ell}{2}$  in front of the hull to  $\frac{\ell}{2}$  aft of the stern and to  $0.6\ell$  in starboard direction (only the half hull is considered) and it contains  $321 \times 121 \times 45$  mesh points in the  $x$ -,  $y$ - and  $z$ -direction, respectively. The hull geometry is discretized with  $162 \times 45$  grid points and is shown in Figure 4.4.

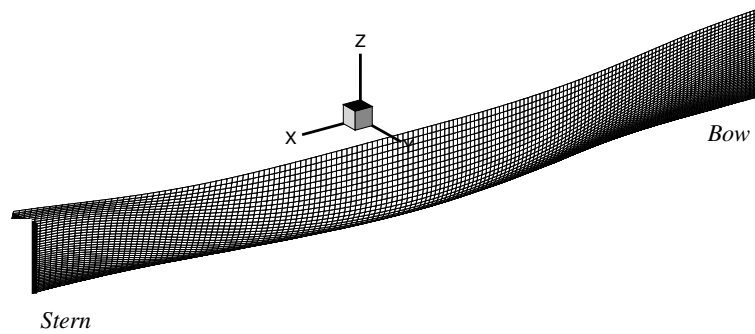


Figure 4.4: Discrete representation of the half hull.

To reduce reflection from the far-field boundaries,  $\Omega_h$  is extended with a dissipation zone as described in Section 2.4.2. The discretized computational domain, including the dissipation zone, is shown in Figure 4.5 and is denoted by  $\hat{\Omega}_h$ . The total number of grid points in  $\hat{\Omega}_h$  is  $353 \times 127 \times 45$ .

As a reference length we take the ship's length,  $\ell = 1$ .

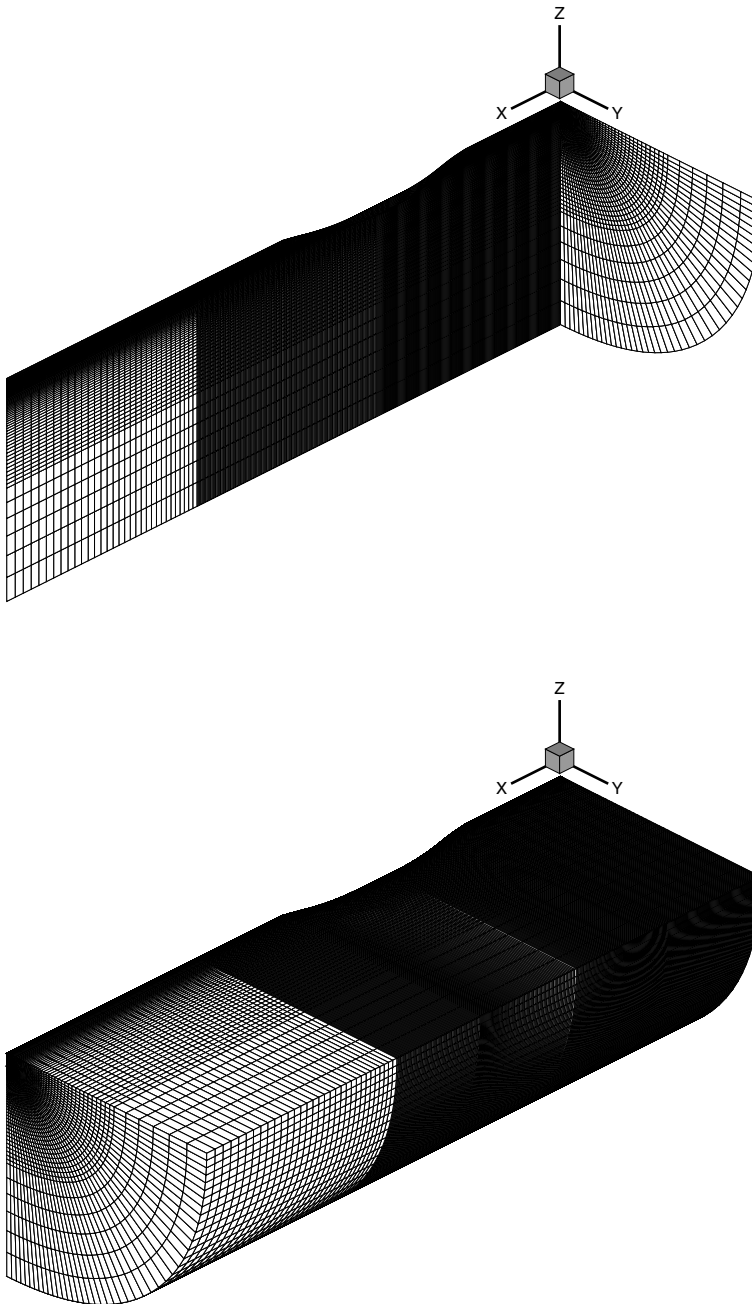


Figure 4.5: The boundaries of the discretized computational domain  $\hat{\Omega}_h$ . *Top figure*: the inflow plane and the plane of symmetry. *Bottom figure*: the outflow plane, the free surface and the external plane.

For all computations performed in this chapter, we take  $(\mathbf{u}, \varphi) = (1, 0, 0, 0)^T$  as initial solution for the iteration process and  $x^3 = 0$  as initial estimate of the free surface. It has to be noted here that this initial mesh does not contain any grid point above the free surface. For the nonlinear free-surface computation the geometry of the hull above the  $x^3 = 0$ -plane is simply extrapolated in the direction of the positive  $x^3$ -axis. As a result the subsequent meshes during the free-surface iteration may be non-smooth at  $x^3 = 0$ , as sketched in Figure 4.6.

The convergence criterion for the preconditioned GMRES iteration method is set at  $2 \times 10^{-3}$ . The (nonlinear) iteration process is continued until  $\|\varphi^n - \varphi^{n-1}\|_\infty \leq 5 \times 10^{-4}$  where  $n$  is the number of nonlinear iterations.

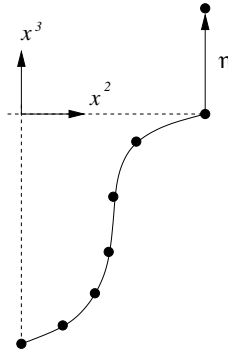


Figure 4.6: Sketch of the non-smooth extension of the mesh occurring during the nonlinear free-surface updates.

### 4.3.1 Two approximations of free-surface flow

To handle the nonlinear quasi free-surface condition (2.7), two approaches can be followed. Either some form of approximation of the free-surface flow problem can be applied, still maintaining the nonlinearity of the underlying (bulk) flow problem, or a fully nonlinear approach. The approximation approach is addressed in this section. In case of approximation several forms are conceivable, e.g., approximation by means of a ‘double-body’ solution or linearization around a uniform (horizontal) flow. A comprehensive study of the various forms of linearization of the free-surface conditions, for irrotational-flow methods, is given in [60].

#### ‘Double-body’ approximation

The ‘double-body’ approximation is most likely to be valid for low Froude numbers. The corresponding theory is named *slow-ship theory*. This theory assumes that the free-surface flow can be decomposed into the flow around the hull having a flat water surface, hence its name, and a small perturbation. The boundary conditions imposed on the free surface ( $x^3 = 0$ ) in the ‘double-body’ approximation are

$$\frac{\partial u^1}{\partial \xi^3} = 0, \quad \frac{\partial u^2}{\partial \xi^3} = 0, \quad u^3 = 0. \quad (4.23)$$

As an example, we have computed the pressure distribution on the free surface belonging to the 'double-body' solution for the Series 60 hull at  $Fr = 0.316$  and  $Re = 10^6$ , see Figure 4.7. Note the absence of the Kelvin wedge. This is due to the symmetry conditions (4.23) imposed at  $x^3 = 0$ .

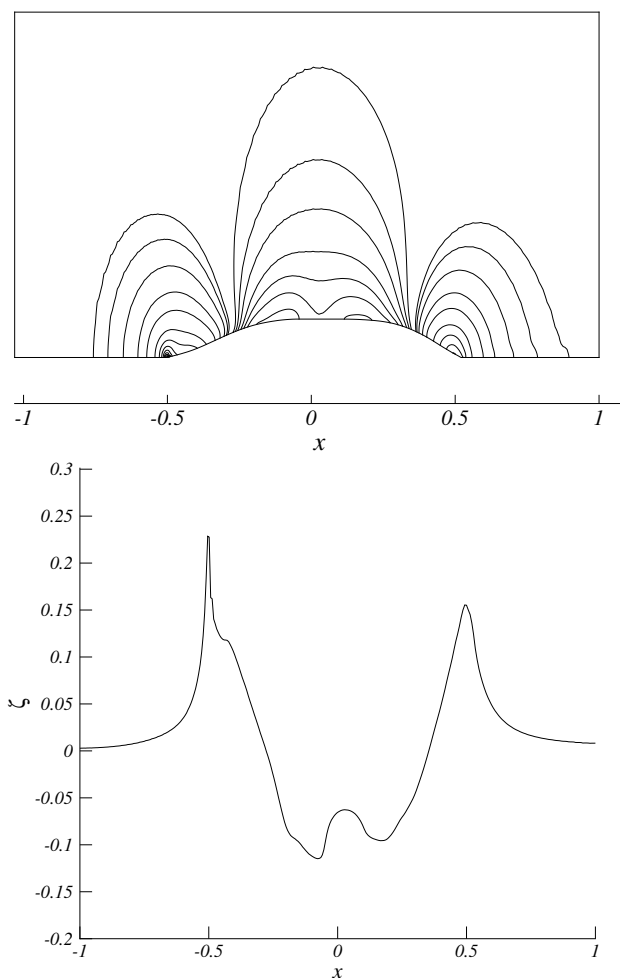


Figure 4.7: Computed pressure distribution (*top figure*) and corresponding waterline in the plane of 'symmetry-and-hull' (*bottom figure*) corresponding to the 'double-body' solution. The hull is located at  $-0.5 \leq x \leq 0.5$ .

To obtain some insight into the development of the viscous regions, i.e., the boundary layer and the wake, we have plotted the iso-contours of the axial velocity component for some cross-sections of  $\Omega_h$  in Figure 4.8.

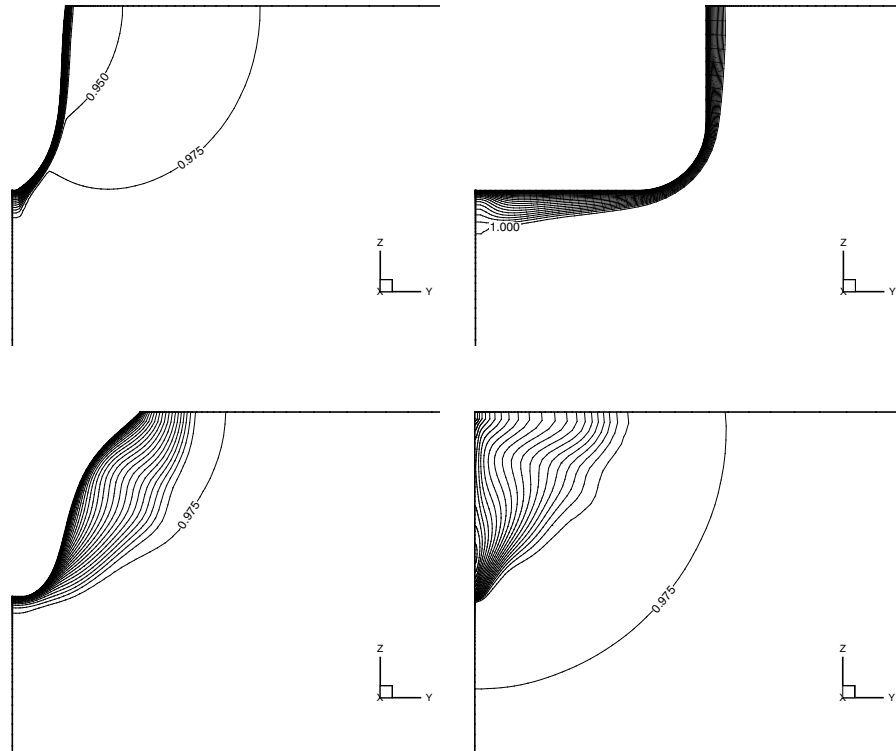


Figure 4.8: Computed  $u^1$ -contours, for the 'double-body' calculation, in cross-sectional planes:  $x = -0.4$  (*top left*),  $x = 0$  (*top right*),  $x = +0.4$  (*bottom left*) and  $x = +0.54$  (*bottom right*). Contour-line increment is 0.025.

The boundary layer clearly appears to grow very rapidly due to the adverse pressure gradient as the cross-sectional area decreases towards the rear of the hull. The convergence history for the equations is shown in Figure 4.9.



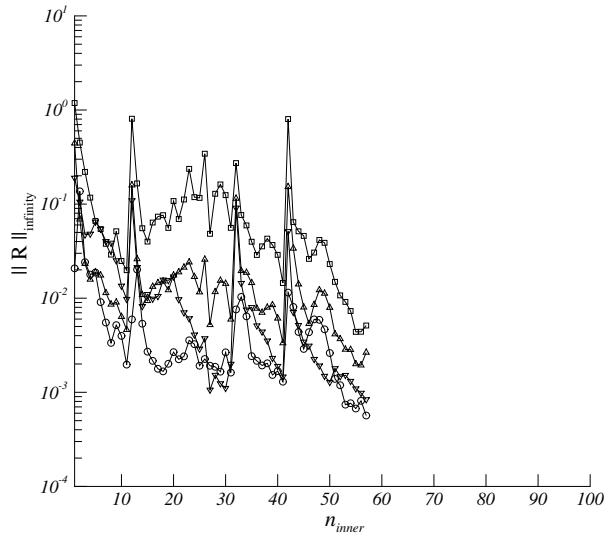


Figure 4.9: Convergence history of the inner-iteration process,  $\mathbf{R}$  is the residual of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ), the  $y$ -momentum equation ( $\circ$ ), and the  $z$ -momentum equation ( $\nabla$ ), measured in  $L_\infty$ -norm.

The large jumps in the residuals are due to the application of the grid-sequencing strategy.

#### 'Uniform-flow' linearization

The uniform-flow type of linearization presupposes the existence of the decomposition

$$\mathbf{u}(\mathbf{x}) = \mathbf{U} + \epsilon \mathbf{u}'(\mathbf{x}) + \mathcal{O}(\epsilon^2), \quad (4.24a)$$

$$\varphi(\mathbf{x}) = \Phi + \epsilon \varphi'(\mathbf{x}) + \mathcal{O}(\epsilon^2), \quad (4.24b)$$

as  $\epsilon \rightarrow 0$ . The perturbation parameter  $\epsilon$  could represent  $\text{Fr}^2$ , since the case  $\text{Fr}^2 = 0$  constitutes no free-surface waves. Here  $\mathbf{U} = (U_\infty, 0, 0)^T$  and  $\Phi = 0$  represent the uniform (horizontal) flow. Of course, this assumption ceases to be valid near the hull. Due to the non-permeability and no-slip conditions, there the perturbation velocity is  $\mathcal{O}(1)$ . We assume that outside the boundary layer, the wake and the stagnation region, the decomposition (4.24) holds. The linearization (4.24) automatically implies a decomposition of the free surface as

$$\eta(x^1, x^2) = \eta_0 + \epsilon \eta'(x^1, x^2), \quad (4.25)$$

with  $\eta_0 = 0$  corresponding to the undisturbed flow and with  $\|\eta'\| \ll 1$ . Application of this linearization to our free-surface method results in imposing, at  $x^3 = \eta_0$ , the

linearized form of the quasi free-surface condition which reads in mixed contravariant/Cartesian velocity components:

$$U_\infty \frac{\partial \varphi}{\partial \xi^1} - \text{Fr}^{-2} u^3 = 0. \quad (4.26)$$

The other boundary conditions imposed on the free surface are, physically seen, the same as the first two in (4.23):

$$\tau^{13} = 0 \Rightarrow \frac{\partial U^1}{\partial \xi^3} = 0 \quad \text{and} \quad \tau^{23} = 0 \Rightarrow \frac{\partial U^2}{\partial \xi^3} = 0, \quad (4.27)$$

where  $U^1$  and  $U^2$  are the contravariant velocity components.

In our numerical computations, in (4.26) we have replaced  $U_\infty$  by  $\bar{U}^1$ , i.e., the local contravariant velocity component, in the  $\xi^1$ -direction, from the previous iteration. This corresponds to a Picard-type linearization of the incomplete quasi free-surface condition (incomplete because of the absence of the  $U^2 \frac{\partial \varphi}{\partial \xi^2} + U^3 \frac{\partial \varphi}{\partial \xi^3}$  term). The discretized quasi free-surface condition yields

$$\bar{U}_j^1 D_{\xi^1}^- \varphi_j - \text{Fr}^{-2} u_j^3 = 0, \quad (4.28)$$

where  $D_{\xi^1}^-$  corresponds to the  $\mathcal{O}(h)$ -upwind scheme.

The computed wave elevation  $\zeta$  is shown in Figure 4.10. As opposed to the ‘double-body’ solution presented in Figure 4.7, the present computed wave pattern already clearly reveals the presence of the Kelvin wedge. So, notice the strong influence of only replacing the kinematic boundary condition  $U^3 = 0$  at the water surface by the quasi-free surface condition: *no Kelvin wedge versus a Kelvin wedge!* Further downstream, and particularly more outboard, the solution still seems to lack the necessary accuracy when making comparisons with, e.g., the free-surface Navier-Stokes results presented in [23], see Figure 4.11.

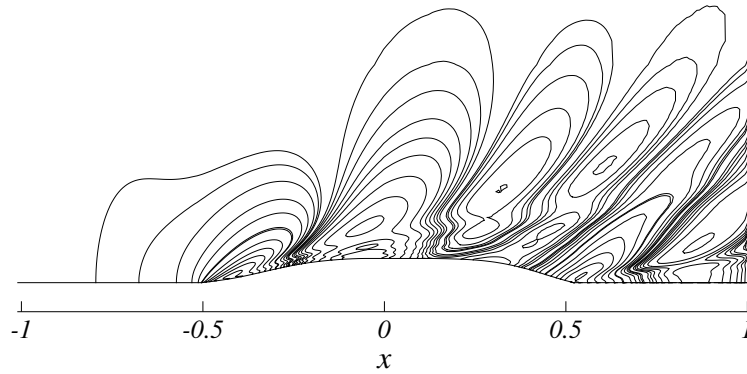


Figure 4.10: Computed wave pattern  $\zeta$  corresponding to the uniform-flow linearization.

A further comparison of the computed wave pattern with results from measurements, taken from [74], is made in Figure 4.12. This comparison shows a fairly good agreement for the wave cut at  $\frac{y}{\ell} = 0.0755$ . Further away from the hull, the amplitude of the computed wave elevation seems to be somewhat too small. This may be attributed to the linearization and to the numerical damping due to the first-order discretization of (4.26).

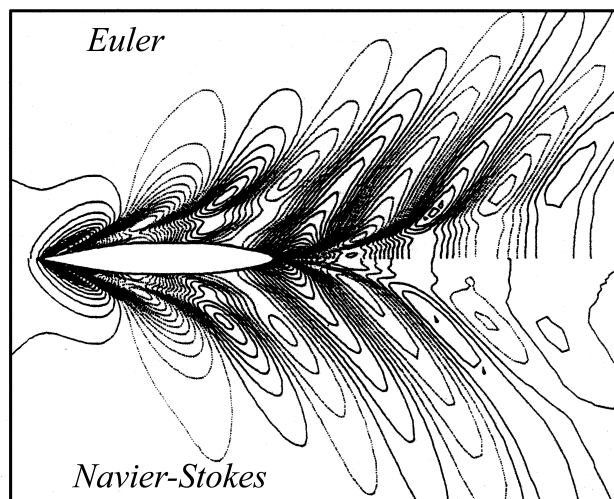


Figure 4.11: Copy of the computed viscous and inviscid wave pattern taken from [23].

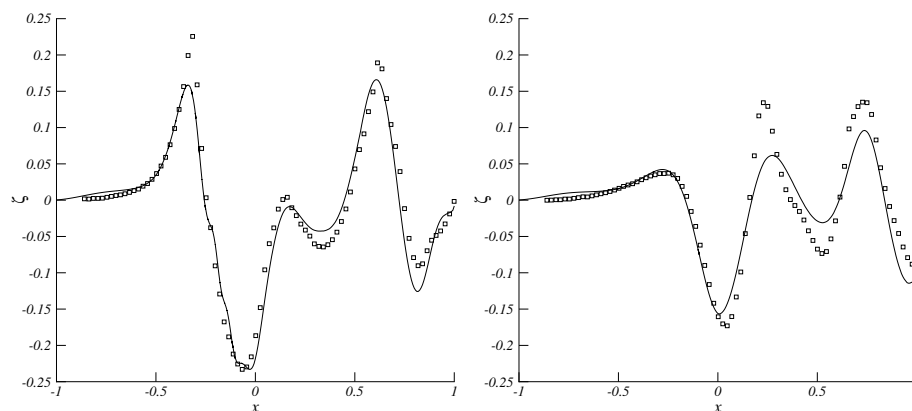


Figure 4.12: Comparison of the computed longitudinal wave cuts (—) with experimental results (□). Left figure:  $\frac{y}{\ell} = 0.0755$ , right figure:  $\frac{y}{\ell} = 0.2067$ ,  $\frac{y}{\ell} = 0$  corresponds to the plane of symmetry.

Figure 4.13 reveals some insight into the deviation of the solution from the uniform flow. This figure shows the deviation of the magnitude of the velocity from the uniform flow velocity  $U_\infty$ , i.e.,

$$\Delta := \left| \frac{\|\mathbf{u}\|_2 - U_\infty}{U_\infty} \right|, \quad (4.29)$$

where  $\|\mathbf{u}\|_2 = (\mathbf{u}, \mathbf{u})^{\frac{1}{2}}$ . As expected, the major deviations from the uniform flow occur in the boundary layer, wake and the bow and stern region. Outside these regions the deviation is very small, thus confirming the validity of the linearization there.

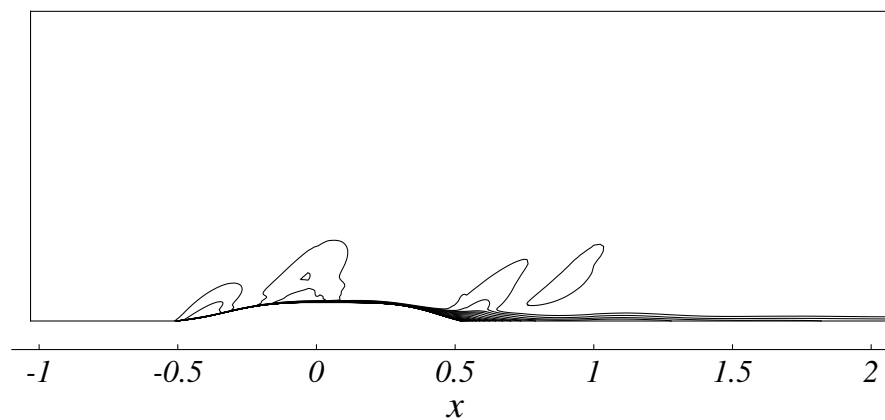


Figure 4.13: Computed spatial distribution of  $\Delta$  on the free surface. The hull is located at  $-\frac{1}{2} \leq x \leq \frac{1}{2}$ .

The iso-contours of the axial-velocity component are plotted in Figure 4.14.

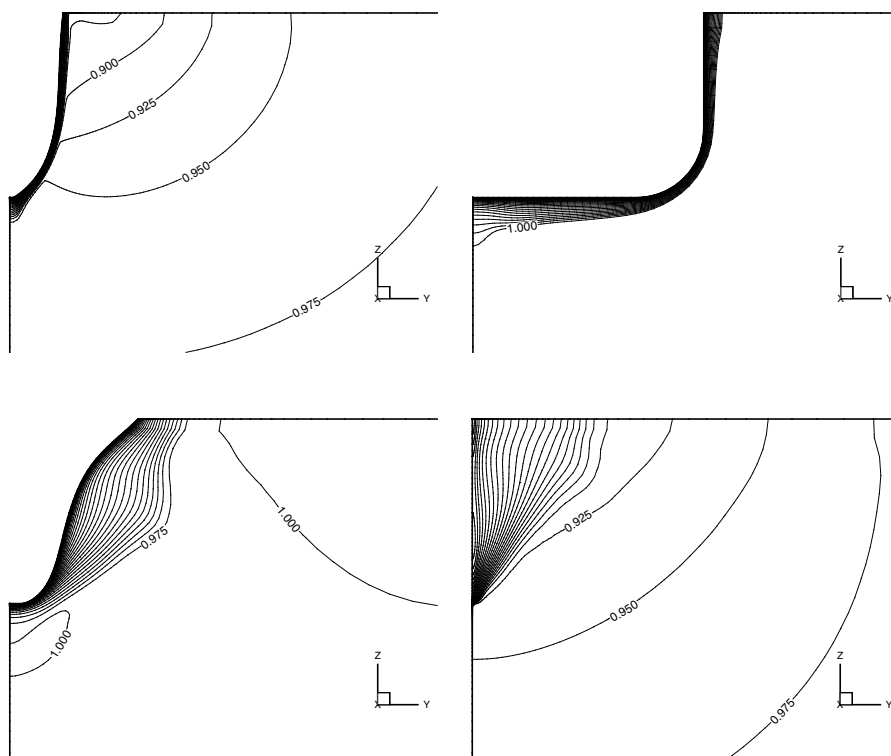


Figure 4.14: Computed  $u^1$ -contours, for the uniform-flow linearization, in cross-sectional planes:  $x = -0.4$  (top left),  $x = 0$  (top right),  $x = +0.4$  (bottom left) and  $x = +0.54$  (bottom right). Contour-line increment is 0.025.

Comparing Figure 4.14 with 4.8 shows the effect of replacing the symmetry conditions, imposed on  $x^3 = 0$ , by the (linearized) quasi free-surface condition. The fact that at  $x = 0$  (Figure 4.14, top right), the boundary layer is thicker at the free surface than at some distance below it, must be caused by a pressure gradient in the  $x$ -direction which is larger at the free surface than below it.

The convergence history for the equations is shown in Figure 4.15. The large jumps in the residuals of the equations are again due to the grid-sequencing strategy which was employed. The computation on the finest grid starts at  $n_{inner} = 150$  and the residuals converge in less than 100 iterations to below the specified tolerance. Comparing Figure 4.15 with Figure 4.9 shows that the (qualitatively much better) solution of the 'uniform-flow' linearization is obtained at a significantly (4 times) higher computational cost than that obtained with the 'double-body' approximation.

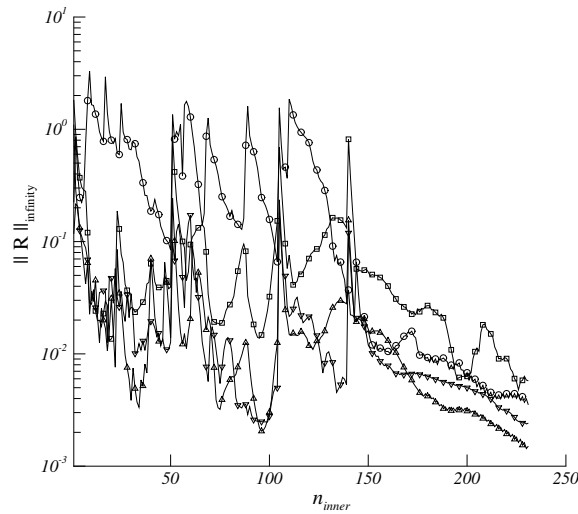


Figure 4.15: Convergence history, for the ‘uniform-flow’ linearization, of the inner-iteration process,  $\mathbf{R}$  is the residual of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ), the  $y$ -momentum equation ( $\circ$ ), and the  $z$ -momentum equation ( $\nabla$ ), measured in  $L_\infty$ -norm.

As already mentioned in Section 3.3.3, our new free-surface iteration method does not suffer from conflicting boundary conditions at the intersection of the free surface and the no-slip boundary. Hence, in the numerical algorithm the quasi free-surface condition can be imposed over the entire free surface. Therefore, of particular interest is the interaction of the boundary layer and wake with the free surface. The current algorithm reveals the presence of waves with high wave numbers in transverse direction in this region, as can already be observed in Figure 4.10 and in more detail in Figure 4.16. These short waves originate at the bow and develop in streamwise direction until the stern is reached. At the stern they have damped out. The grid points, as marked in Figure 4.17, clearly show that these waves which are short in transverse direction (in the remainder briefly called: short waves) are not ‘odd-even grid-point waves’. We think that the short waves are physically correct. More widely interpreting Lighthill’s theory about surface gravity-waves for water of uniform depth (Section 3.3 in [50]), our conjecture is that inside the boundary layer around the hull, at local flow speeds lower than the far-field flow speed, short stationary surface gravity-waves (which also satisfy the dispersion relation relating flow speed and wavelength) can co-exist with the long stationary wave corresponding with the far-field flow speed. To our knowledge, the short waves have not been observed yet in published free-surface Navier-Stokes flow computations for this Series 60 test case.

As mentioned, when the stern is reached the short waves have already disappeared. Furthermore, they very quickly damp in vertical direction (negative  $z$ -direction). Their amplitudes decay as  $e^{kz}$ , where  $k$  is the local wave number (see Section 3.2 in [50]). (The zero-stress conditions imposed at the free surface are

responsible for the generation of this free-surface boundary layer.) In the current mesh the minimum distance of the first layer of grid points underneath the free surface is approximately  $2 \times 10^{-3}$ . So any free-surface wave with a half wavelength smaller than this is not noticed anymore in the first grid-point layer underneath the free surface. The free-surface boundary layer cannot be resolved on the current mesh due to insufficient resolution in the  $x^3$ -direction.

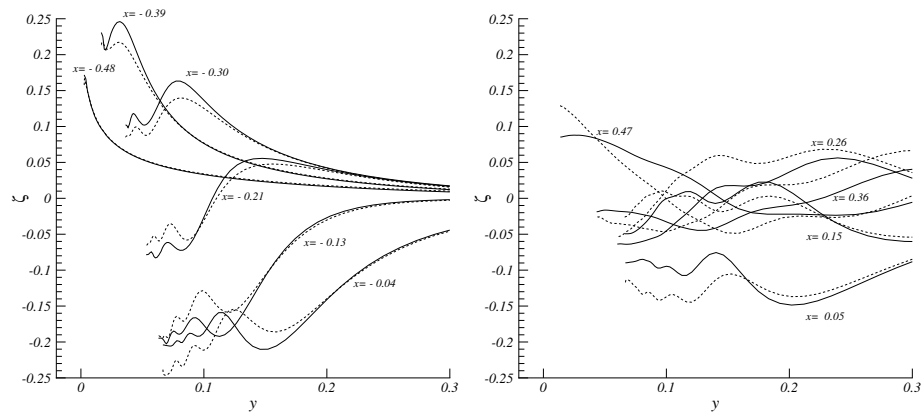


Figure 4.16: Computed wave elevation  $\zeta$  at the free surface (solid) and at the first layer of nodes underneath the free surface (dashed) for  $x < 0$  (left figure) and  $x > 0$  (right figure).

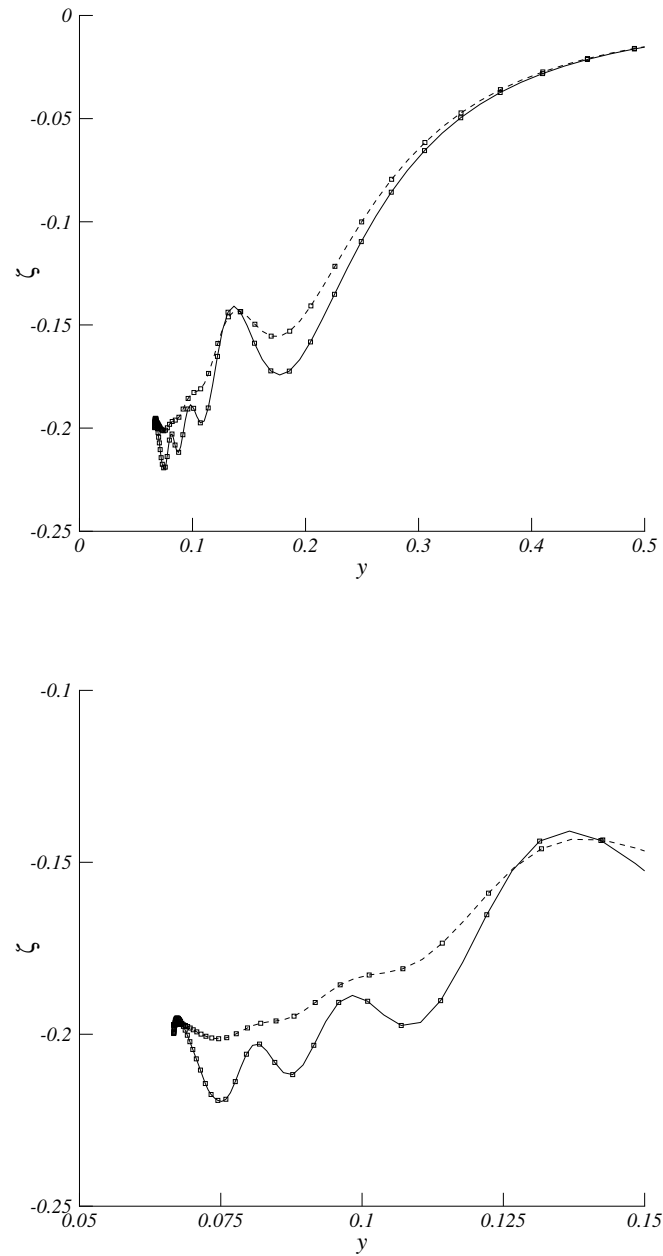


Figure 4.17: Computed wave elevation  $\zeta$  at the free surface (solid) and at the first layer of nodes underneath the free surface (dashed) for  $x = 0$  (*top figure*) and a magnification of the two curves in the region near the hull (*bottom figure*). Only every second marker is shown.



Concerning the question to what extent these short waves in the free-surface boundary layer contribute to a ship's total drag, the conjecture is that this is very little, in general. As such, trying to resolve them more accurately does not seem to be necessary.

#### 4.3.2 Nonlinear free-surface method with remeshing

So far, our remeshing algorithm, described in Section 4.2.3, has not yet been applied. We will do so in the present section. The short waves have a negative effect on the quality of the new grids generated by our remeshing algorithm. The short waves appear to lead to grids which suffer from mesh tangling, precisely in the short-wave region. Our provisional remedy against this is to resolve the waves less accurately in transverse direction by taking a somewhat coarser mesh in the  $x^2$ -direction. This may go at the expense of the resolution of the boundary layer and wake. Another remedy would be to explicitly damp the short waves, which is not trivial for the general case, given the presence of the long waves. A genuine remedy would be to make the remeshing algorithm robust against these short waves. This could be achieved by applying the elliptic grid generator, used for the initial grid, for every new grid.

An approximation of the nonlinear free-surface solution is computed with the aid of the modified free-surface algorithm, as described in Section 2.5.1. The coarsened mesh is extracted from  $\hat{\Omega}_h$  by specifying the minimum mesh width in  $x^2$ -direction to satisfy  $\Delta x^2 \geq 1.25 \times 10^{-3}$ . This results in a mesh with 69 instead of 127 grid points in  $x^2$ -direction. The complete quasi free-surface condition is imposed on the approximate locations of the free surface during the iteration process, and the mesh is adapted to the new free-surface location, following the remeshing algorithm proposed in Section 4.2.3. Here, in the solution process, (4.14) is treated using Newton's method and the pressure gradient  $\nabla\varphi$  is discretized employing the  $\mathcal{O}(h)$ -upwind scheme, yielding

$$\bar{U}_j^\alpha D_{\xi^\alpha}^- \varphi_j + U_j^\alpha D_{\xi^\alpha}^- \bar{\varphi}_j - \bar{U}_j^\alpha D_{\xi^\alpha}^- \bar{\varphi}_j - \text{Fr}^{-2} u_j^3 = 0, \quad (\text{sum over } \alpha), \quad (4.30)$$

where  $D_{\xi^\alpha}^- = E_{\xi^\alpha}^0 - E_{\xi^\alpha}^{-1}$  and  $\bar{\cdot}$  denotes a value from the previous iteration.

The computed wave pattern obtained with our nonlinear free-surface method is shown in Figure 4.18 and a comparison with the experimental data is made in Figure 4.19.

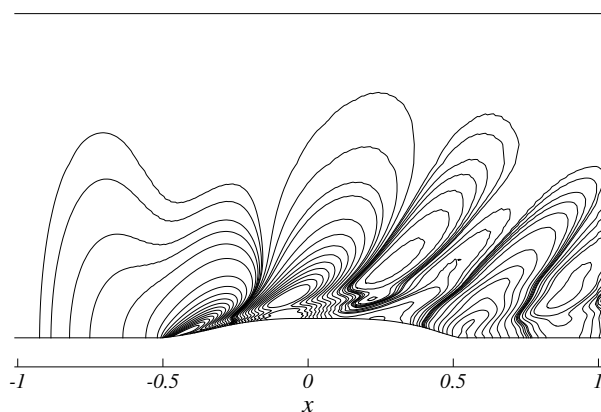


Figure 4.18: Computed wave pattern  $\zeta$  as obtained with our nonlinear free-surface method.

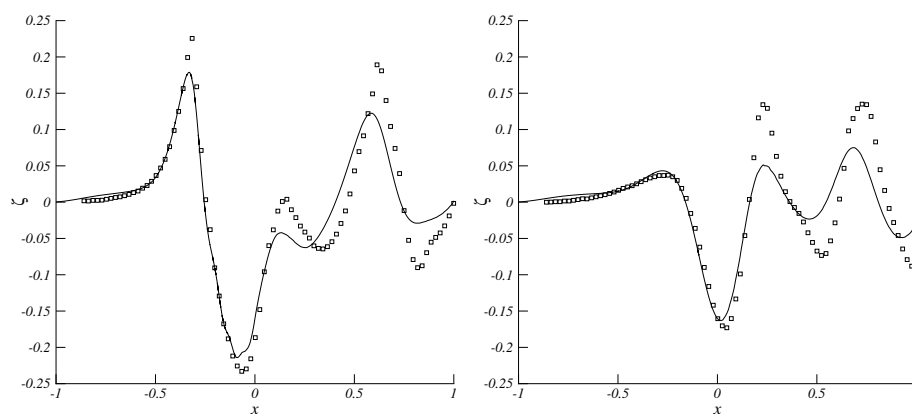


Figure 4.19: Comparison of the computed longitudinal wave cuts (—), obtained with the nonlinear method, with experimental results ( $\square$ ). *Left figure:*  $\frac{y}{\ell} = 0.0755$ , *right figure:*  $\frac{y}{\ell} = 0.2067$ ,  $\frac{y}{\ell} = 0$  corresponds to the plane of symmetry.

A comparison of the Figures 4.18 and 4.10 reveals a difference in the computed wave pattern at the outer edge of the Kelvin wedge. The third extremum, computed with the fully nonlinear algorithm, is less developed. This also clearly appears from comparison of Figure 4.19 and 4.12. Comparison of Figure 4.18 and the Navier-Stokes results in Figure 4.11 shows a fairly good resemblance.

The development of the boundary layer along the hull and in the wake is shown in Figure 4.20.

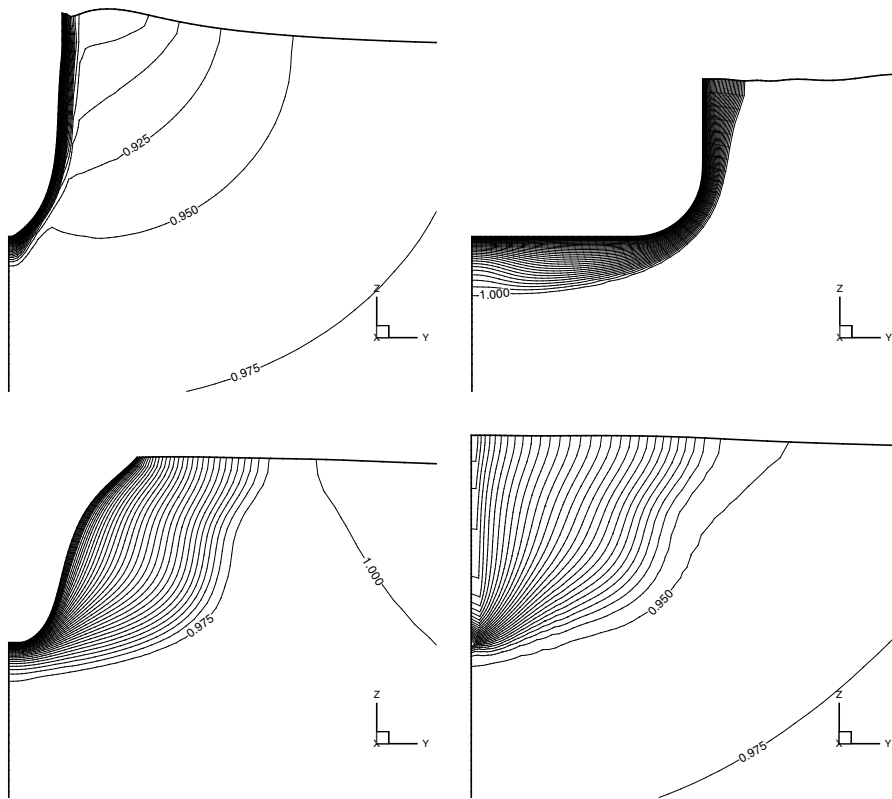


Figure 4.20: Computed  $u^1$ -contours, for the fully nonlinear calculation, in cross-sectional planes:  $x = -0.4$  (*top left*),  $x = 0$  (*top right*),  $x = +0.4$  (*bottom left*) and  $x = +0.54$  (*bottom right*). Contour-line increment is 0.025.

For comparison purposes we repeat the 'uniform-flow' computation on the current grid, i.e., the grid with 69 grid points in the  $y$ -direction. To study the effect of the coarsening on the resolution of the boundary layer we plotted the axial velocity contours obtained with this computation in Figure 4.21.

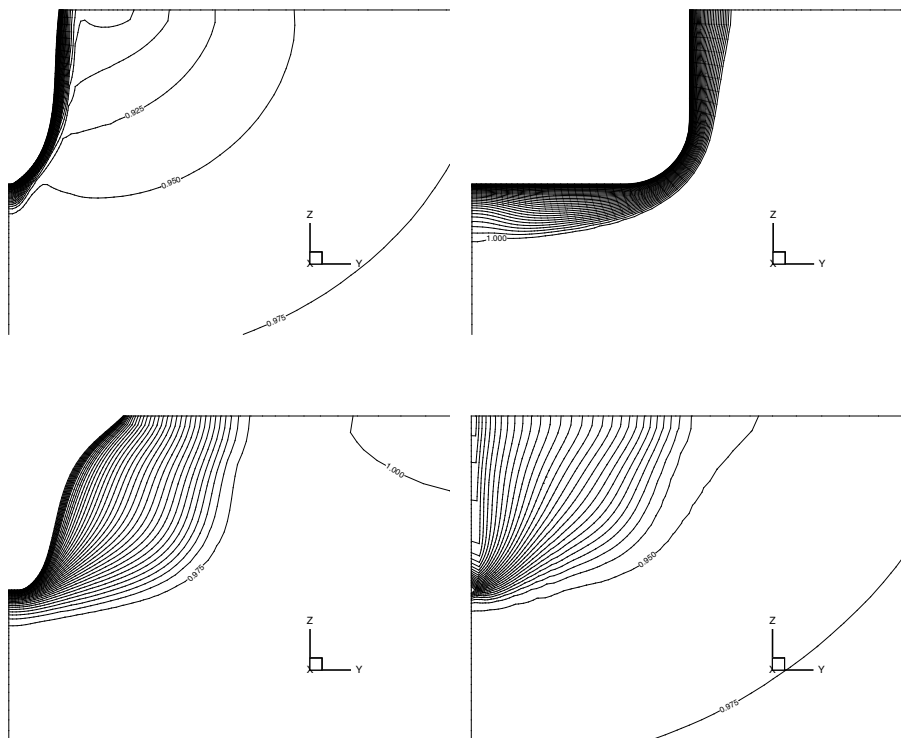


Figure 4.21: Computed  $u^1$ -contours, for the 'uniform-flow' calculation on the coarsened grid, in cross-sectional planes:  $x = -0.4$  (*top left*),  $x = 0$  (*top right*),  $x = +0.4$  (*bottom left*) and  $x = +0.54$  (*bottom right*). Contour-line increment is 0.025.

From a comparison of Figures 4.21 and 4.14 it can be concluded that the thickness of the boundary layer and the wake has increased, with the coarsening of the mesh. Comparing Figure 4.21 with Figure 4.20 reveals that the linearization of the free surface does not have a large effect on the boundary layer. This suggests that for the computation of the viscous drag a linearization of the free surface effect is sufficient. However, for the wave drag this is questionable. Figure 4.22 reveals significant differences between the linearized and nonlinear free-surface shape. Comparison of Figure 4.22 and 4.12 suggests that the coarsening in the  $y$ -direction (69 instead of 127 grid points) has also influenced the wave pattern.

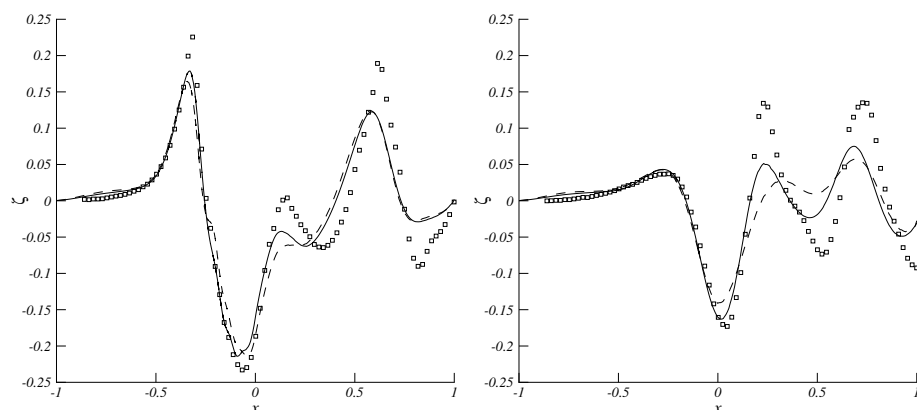


Figure 4.22: Comparison of the computed longitudinal wave cuts, obtained with the nonlinear method (solid) and the 'uniform-flow' linearization method (dashed), with experimental results (markers). *Left figure:*  $\frac{y}{\ell} = 0.0755$ , *right figure:*  $\frac{y}{\ell} = 0.2067$ ,  $\frac{y}{\ell} = 0$  corresponds to the plane of symmetry.

The convergence history for the 'uniform-flow' linearization case on the  $353 \times 69 \times 45$  grid is shown in Figure 4.23.

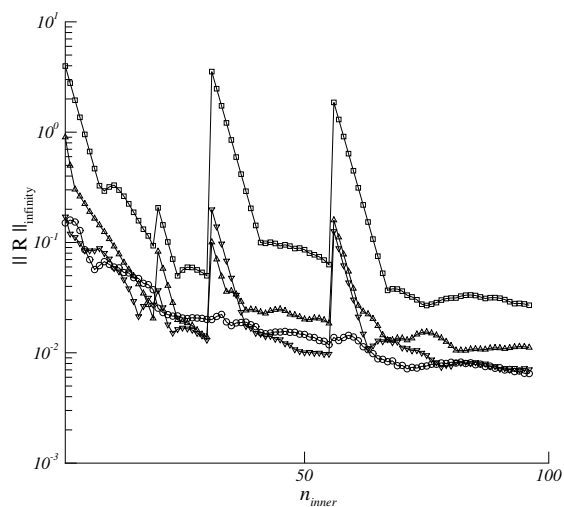


Figure 4.23: Convergence history of the inner-iteration process for the 'uniform-flow' computation on the  $353 \times 69 \times 45$  grid,  $\mathbf{R}$  is the residual of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ), the  $y$ -momentum equation ( $\circ$ ), and the  $z$ -momentum equation ( $\nabla$ ), measured in  $L_\infty$ -norm.

For the nonlinear free-surface computation, the convergence histories for the residuals of the equations and the free-surface pressure defect are still shown in

Figure 4.24. The large jumps in the residuals of the equations are again due to the grid-sequencing strategy. The iteration on the finest grid starts at  $n = 64$ . The modified free-surface algorithm is activated once the finest grid has been reached. The total computational work, measured in the number of nonlinear iterations, is a factor 1.67 larger, for the nonlinear computation, than for the computation with the 'uniform-flow' linearization.

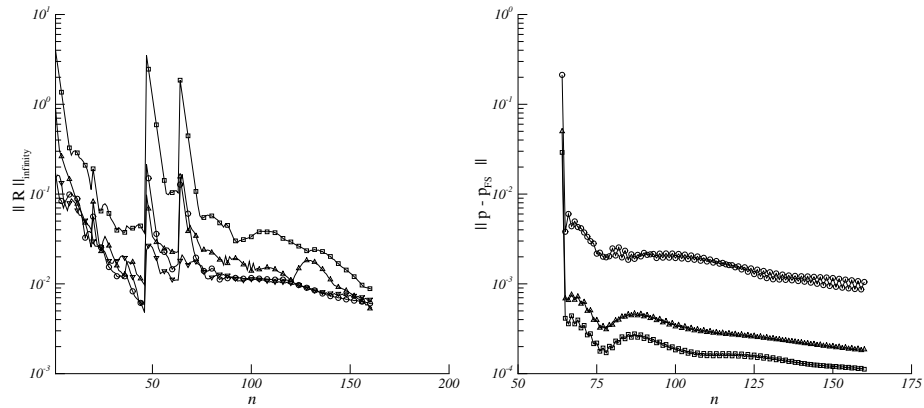


Figure 4.24: Convergence histories for the  $\mathcal{O}(h)$ -upwind discretization of the quasi free-surface condition on the coarsened mesh computed with the modified algorithm. *Left:* of  $\mathbf{R}$ , the residual of the continuity equation ( $\Delta$ ), the  $x$ -momentum equation ( $\square$ ), the  $y$ -momentum equation ( $\circ$ ), and the  $z$ -momentum equation ( $\nabla$ ). *Right:* of the free-surface pressure defect; measured in the  $L_1$ -norm ( $\square$ ),  $L_2$ -norm ( $\Delta$ ), and  $L_\infty$ -norm ( $\circ$ ).

#### 4.4 Conclusions

The topic of this chapter was the application of the new 3D free-surface algorithm, as introduced in the previous two chapters, to a practical application originating from the field of numerical ship hydrodynamics. The test case in this chapter was the computation of the complete flow field, i.e., including the wave pattern, of the Series 60 hull at  $Fr = 0.316$  and  $Re = 10^6$ .

From the numerical results it can be concluded that the mechanism of free-surface flow computation with the 3D quasi-free surface condition imposed on an approximate location of the free surface yields the proper 3D wave physics. It also shows that a fairly accurate estimate of the wave pattern can be obtained from a linearization of the quasi free-surface condition around a uniform flow. (As expected, the 'double-body' approximation yields very inaccurate results for this test case.) In the case of 'uniform-flow' linearization, still no remeshing has to be performed. This method is very suitable for obtaining a first indication of the viscous free-surface flow. A more refined approximation of the wave pattern requires the solution of the fully nonlinear free-surface flow problem. The results presented here suggest that also for ship-hull flows the modified algorithm can significantly reduce the CPU time.

A further benefit of the current method is that no special measures have to be taken to circumvent the incompatibility of the kinematic free-surface condition and the no-slip boundary condition in case of a free-surface penetrating object. The numerical results have shown that in the viscosity-dominated regions a high-wavenumber wave is present, of which the existence is not known from classical analytical theory on water waves. The relative importance of the contribution of this wave to the total drag is expected to be negligible.

The discretization of the quasi free-surface condition as applied in the fully nonlinear free-surface algorithm was first-order accurate. So far, no converged results could be obtained with the second-order discretization of the quasi free-surface condition. It could be that an even larger chance of mesh tangling in case of second-order discretization is one cause of this. In the next chapter we will also formally analyze the existence properties of steady free-surface waves for both the  $\mathcal{O}(h^2)$  and the  $\mathcal{O}(h)$  upwind discretization of the quasi free-surface condition.

To prevent the present remeshing algorithm from generating tangled meshes, the waves which are short in transverse direction (called here the short waves) may need to be damped in transverse direction, or simply resolved less accurately on a coarser grid in transverse direction. Both fixes are acceptable if the short waves are already poorly resolved in vertical direction. The latter may easily happen since, according to theory, the short waves damp out very rapidly in this direction. A difficulty of damping the short waves is to avoid damping the long waves. A drawback of resolving on a coarser grid in transverse direction is that both the viscous phenomena and the wave phenomena are resolved less accurately. A genuine remedy against mesh tangling could be the application of a well-proven (elliptic) grid generator after each free-surface update. We still remark that a capturing method would not have difficulties with short waves.

The main conclusion of this chapter is that the computation of steady ship-hull flows with both free-surface and viscous effects can be done efficiently through the quasi free-surface condition and the modified algorithm.





## CHAPTER 5

# Analysis of Continuous and Semi-Discrete Water-Wave Problem

### 5.1 Introduction

In this chapter we analyze the solution behaviour for the continuous as well as the discretized Navier-Stokes equations, with the quasi free-surface boundary condition imposed at a boundary. The analysis proceeds using classical perturbation methods, linearization and Fourier techniques. We consider the generic problem of perturbations on a uniform horizontal flow in infinitely deep water in two dimensions. In Section 5.2 the analysis is performed for the continuous equations: in Section 5.2.1 for the full Navier-Stokes equations and in Section 5.2.2 for the reduced Navier-Stokes equations as model for the bulk flow. Section 5.3 analyzes the system of modified equations, which results from the semi-discrete reduced Navier-Stokes equations: Section 5.3.1 for a second-order accurate discretization of the quasi free-surface condition and Section 5.3.2 for the first-order upwind discretization of this condition.

### 5.2 Analysis of continuous problem

We consider a linearized model problem similar to the one approximated in Chapter 2, but with the difference that we do it here in  $\mathbb{R}^2$  instead of in  $\mathbb{R}^3$ . The behaviour of free-surface perturbations imposed on the uniform horizontal flow is studied. We assume a perturbation expansion in powers of  $\epsilon$  of the form

$$\mathbf{q}(\mathbf{x}, t; \epsilon) = \mathbf{Q}(\mathbf{x}, t) + \epsilon \mathbf{q}'(\mathbf{x}, t) + \mathcal{O}(\epsilon^2), \quad \text{as } \epsilon \rightarrow 0, \quad (5.1)$$

and

$$\eta(x, t; \epsilon) = H(x, t) + \epsilon \eta'(x, t) + \mathcal{O}(\epsilon^2), \quad \text{as } \epsilon \rightarrow 0, \quad (5.2)$$

where  $\mathbf{q}'(\mathbf{x}, t) = (\mathbf{u}'(\mathbf{x}, t), \varphi'(\mathbf{x}, t))^T$ . The unperturbed flow velocity and pressure, denoted by  $\mathbf{Q}(\mathbf{x}, t) = (\mathbf{U}(\mathbf{x}, t), \Phi(\mathbf{x}, t))^T$ , and the unperturbed water height, denoted by  $H(x, t)$ , are

$$\mathbf{U}(\mathbf{x}, t) = (U, 0)^T, \quad U = \text{constant}, \quad \Phi(\mathbf{x}, t) = 0 \quad \text{and} \quad H(x, t) = 0. \quad (5.3)$$

From definition (2.3) it follows, assuming  $p = 0$  on the free surface  $z = \eta(x, t)$ , that the water height can be directly expressed in terms of the hydrodynamic pressure:

$$\eta(x, t) := \text{Fr}^2 \varphi(x, t). \quad (5.4)$$

Hence, through (5.4) the unknown  $\eta(x, t)$  is effectively removed.

Substitution of the power series (5.1) into the full Navier-Stokes equations and neglecting the second- and higher-order perturbation terms, yields, upon deleting the primes, the following system of linear equations

$$\mathbf{u}_t + \mathbf{U} \cdot \nabla \mathbf{u} + \nabla \varphi - \text{Re}^{-1} \Delta \mathbf{u} = 0, \quad \forall (\mathbf{x}, t) \in \Omega \times \mathcal{T}, \quad (5.5a)$$

$$\nabla \cdot \mathbf{u} = 0, \quad \forall (\mathbf{x}, t) \in \Omega \times \mathcal{T}, \quad (5.5b)$$

which are supplemented with suitable initial and boundary conditions to yield a well-posed problem.

Our spatial domain  $\Omega$  is defined as

$$\Omega = \{\mathbf{x} \in \mathbb{R}^2 : x \in (-\pi, \pi), z \in (\eta, -\infty)\}. \quad (5.6)$$

Since we are considering the flow in infinitely deep water in  $z$ -direction, the domain  $\Omega$  is only bounded by the free surface  $z = \eta(x, t)$ , at which the solution has to satisfy the quasi free-surface condition, which reads in perturbation form:

$$\varphi_t + \mathbf{U} \cdot \nabla \varphi - \text{Fr}^{-2} \mathbf{u} \cdot \mathbf{e}_z = 0, \quad (5.7)$$

and may be imposed on  $z = H(x, t)$  for a consistent linearization upto  $\mathcal{O}(\epsilon^2)$ . For  $z \downarrow -\infty$  the uniform flow solution is imposed. Furthermore, we assume periodicity of the solution  $\mathbf{q} := (\mathbf{u}, \varphi)^T$  in  $x$ -direction, i.e.,

$$\mathbf{q}(-\pi, z) = \mathbf{q}(\pi, z). \quad (5.8)$$

### 5.2.1 Fourier analysis of full Navier-Stokes equations

To construct a Fourier representation of the solution  $\mathbf{q}(\mathbf{x}, t)$  consider the following isolated mode

$$\mathbf{q}(\mathbf{x}, t) = \hat{\mathbf{q}}(k, s, \omega) e^{ikx + sz - i\omega t}, \quad (5.9)$$

where  $k, s \in \mathbb{R}$  are the wave numbers in  $x$ - and  $z$ -direction, respectively, and where  $\omega \in \mathbb{R}$  is the radial frequency. Substitution of (5.9) into (5.5) yields

$$\hat{\mathbf{L}}(k, s, \omega) \hat{\mathbf{q}}(k, s, \omega) e^{ikx + sz - i\omega t} = \mathbf{0}, \quad (5.10)$$

with  $\hat{\mathbf{L}}(k, s, \omega)$  representing the Fourier symbol of (5.5):

$$\hat{\mathbf{L}}(k, s, \omega) := \begin{pmatrix} \hat{H}(k, s, \omega) & 0 & \hat{G}_x(k, s) \\ 0 & \hat{H}(k, s, \omega) & \hat{G}_z(k, s) \\ \hat{D}_x(k, s) & \hat{D}_z(k, s) & 0 \end{pmatrix}, \quad (5.11)$$

where  $\hat{H}(k, s, \omega) := -i\omega + ikU - \text{Re}^{-1}(s^2 - k^2)$  is the Fourier symbol of the convection-diffusion operator,  $\hat{\mathbf{G}}(k, s) = (ik, s)^T$  the Fourier symbol of the pressure gradient and  $\hat{\mathbf{D}}(k, s) = (ik, s)^T$  the Fourier symbol of the divergence operator  $\text{div}\mathbf{u}$ . A nontrivial solution of (5.10) exists only if

$$\hat{\mathbf{q}}(k, s, \omega) \in \mathcal{N}(\hat{\mathbf{L}}(k, s, \omega)) \Leftrightarrow \det(\hat{\mathbf{L}}(k, s, \omega)) = 0, \quad (5.12)$$

where  $\mathcal{N}$  represents the nullspace. Expanding the determinant condition yields

$$\hat{H}(k, s, \omega) \left( \hat{G}_x \hat{D}_x + \hat{G}_z \hat{D}_z \right) = 0 \Leftrightarrow \hat{H}(k, s, \omega)(k^2 - s^2) = 0. \quad (5.13)$$

The various roots of (5.13) generate the following bases of  $\mathcal{N}(\hat{\mathbf{L}}(k, s))$ :

$$\text{if } \omega, k, s = 0, \quad \text{span} \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\} \quad (5.14a)$$

and otherwise

$$\text{if } k^2 - s^2 = 0, \quad \text{span} \left\{ \begin{pmatrix} \hat{G}_x(k) \\ \hat{G}_z(k) \\ -\hat{H}(k, \omega) \end{pmatrix} \right\}, \quad (5.14b)$$

$$\text{if } \hat{H}(k, s, \omega) = 0, \quad \text{span} \left\{ \begin{pmatrix} \hat{G}_z(k) \\ -\hat{G}_x(k) \\ 0 \end{pmatrix} \right\}. \quad (5.14c)$$

The space spanned by (5.14a) corresponds to *constant modes*. These are not of interest to us. The spaces spanned by (5.14b) and (5.14c) are of interest and correspond to the *inviscid modes* and the *viscous modes*, respectively. This terminology has been adopted from Chapter 3 of [12]. Note that the condition  $k^2 - s^2 = 0$  removes the viscous contribution from  $\hat{H}(k, s, \omega)$ . Considering only the inviscid mode, the solution can be written as

$$\mathbf{q}(x, z, t) = \begin{pmatrix} \hat{G}_x(k) \\ \hat{G}_z(k) \\ -\hat{H}(k, \omega) \end{pmatrix} e^{ikx+kz-i\omega t}, \quad (5.15)$$

where we have used the fact that for

$$\lim_{z \downarrow -\infty} \mathbf{u}(\mathbf{x}, t) = 0 \Rightarrow s = +k, \quad k \in \mathbb{R}^+. \quad (5.16)$$

This solution also has to satisfy the quasi free-surface boundary condition (5.7) on  $z = H(x, t)$ . This yields the following condition on  $k$  and  $\omega$

$$i\omega \hat{H}(k, \omega) - iUk\hat{H}(k, \omega) - \text{Fr}^{-2}k = 0. \quad (5.17)$$

Condition (5.17) is known as the *dispersion relation*. Here the roots  $\omega_{1,2}$  of the dispersion relation can be calculated explicitly

$$\omega_{1,2} = Uk \pm \sqrt{\text{Fr}^{-2}k}. \quad (5.18)$$

If (5.18) is substituted into (5.9), it follows that an isolated Fourier mode behaves as

$$e^{ikx - i(Uk \pm \sqrt{\text{Fr}^{-2}k})t}. \quad (5.19)$$

These waves are known as *surface gravity waves*. They move with *phase velocity*

$$c(k) = U \pm \sqrt{\text{Fr}^{-2}k^{-1}}. \quad (5.20)$$

Note that for  $U = 0$ , i.e., moving the frame of reference with the flow, the familiar dispersion relation for waves propagating in still water is obtained, [80]. Also the wave energy of the gravity waves travels with the *group velocity*, which is defined as

$$\frac{d\omega}{dk} = U \pm \frac{1}{2} \sqrt{\text{Fr}^{-2}k^{-1}}. \quad (5.21)$$

For an observer traveling with the flow velocity  $U$  the group velocity is exactly half the phase velocity. This means that the observer traveling with the flow velocity, will always see waves with single wave number  $k$  and frequency  $\omega(k)$ , as opposed to an observer following a particular crest, who will see waves with different wave numbers  $k$ .

Since we are only interested in the stationary problem the dispersion relation is determined by (5.18) for  $\omega_{1,2} = 0$ . This yields

$$U^2k = \text{Fr}^{-2}. \quad (5.22)$$

This relation determines the wave number  $k$  of waves which can keep a steady position against the oncoming stream. This complies with the result already derived in [50]. The dispersion relation (5.22) can also directly be obtained from a *steady formulation* of the initial boundary-value problem (5.5) – (5.7). Both formulations, the steady and the unsteady for  $t \rightarrow \infty$ , lead to a wave solution satisfying dispersion relation (5.22).

### 5.2.2 Fourier analysis of reduced Navier-Stokes equations

Now we will analyze the effects of neglecting the streamwise diffusion in the Navier-Stokes operator. Hence, the Laplacian in (5.5) is replaced by

$$\tilde{\Delta} := \partial_z^2, \quad \text{in } \mathbb{R}^2. \quad (5.23)$$

Following the same analysis as above, the Fourier symbol of the convection-diffusion operator is now defined as

$$\hat{H}(k, \omega) := -i\omega + ikU - \text{Re}^{-1}k^2, \quad (5.24)$$

where we have used condition (5.14b). Observe that, for the reduced Navier-Stokes equations, the condition (5.14b) does not remove the viscosity completely. Substitution of (5.15), with  $\hat{H}(k, s, \omega)$  defined as (5.24), into (5.17) yields the dispersion relation for the reduced Navier-Stokes problem. Also in this case the roots  $\omega_{1,2}$  can be calculated explicitly, giving

$$\omega_{1,2} = Uk \pm \frac{1}{2} \sqrt{4\text{Fr}^{-2}k - \text{Re}^{-2}k^4} + i\frac{1}{2}\text{Re}^{-1}k^2. \quad (5.25)$$

Hence, if (5.25) is inserted in (5.9), it follows that a surface gravity wave for the reduced system behaves as

$$e^{\frac{1}{2}\text{Re}^{-1}k^2 t} e^{ikx - i(Uk \pm \frac{1}{2}\sqrt{4\text{Fr}^{-2}k - \text{Re}^{-2}k^4})t}. \quad (5.26)$$

Only when neglecting the viscous terms completely, this reduced-system surface gravity wave moves with the same phase and group velocity as in the full Navier-Stokes case, i.e., with the velocities (5.20) and (5.21), respectively. Following the same line of reasoning as for the full Navier-Stokes equations, shows that no steady waves can exist for finite  $\text{Re}$ . (Note that the possibility of a negative sign of the term under the square root ( $4\text{Fr}^{-2}k - \text{Re}^{-2}k^4$ ) does not alter this.) The steady waves that are possible for the full Navier-Stokes equations, i.e., which satisfy the dispersion relation (5.22), reveal an exponential growth in time in case of the reduced Navier-Stokes equations, a growth which is proportional to  $\text{Re}^{-1}$  and  $k^2$ . This suggests that this formulation of the free-surface flow problem is formally ill-posed. However, the measure of ill-posedness is small for large  $\text{Re}$ .

The equations considered in the preceding chapters are all in steady form. Repeating the analysis for the steady formulation of the equations and postulating a solution of the form  $e^{ikx+sz}$  with  $k, s \in \mathbb{R}$ , yields the following expressions for the convection-diffusion operator

$$\hat{H}(k) := ikU - \text{Re}^{-1}k^2, \quad (5.27)$$

where we have again used  $s^2 = k^2$  ( $\Rightarrow s = +k$ ), and the quasi free-surface condition

$$-ikU\hat{H}(k) - \text{Fr}^{-2}k = 0 \quad \Leftrightarrow \quad i\text{Re}^{-1}Uk^3 + U^2k^2 - \text{Fr}^{-2}k = 0. \quad (5.28)$$

From (5.28) it is immediately clear that the inviscid limit yields the familiar dispersion relation (5.22). Note that (5.28) has no real solutions other than the trivial solution  $k = 0$ . As expected, this steady analysis yields the same outcome as the foregoing unsteady analysis. If one would allow  $k, s \in \mathbb{C}$  then the solutions of the cubic equation (5.28) would be

$$\begin{aligned} k_1 &= 0, \quad \text{and} \quad k_{2,3} := \xi + i\zeta, \quad \text{with} \\ \xi &= \frac{1}{4}\text{Re}U \sqrt{2 \left( \sqrt{1 + 16U^{-6}\text{Re}^{-2}\text{Fr}^{-4}} - 1 \right)}, \\ \zeta &= \frac{1}{2}\text{Re}U \left( 1 \pm \frac{1}{2} \sqrt{2 \left( \sqrt{1 + 16U^{-6}\text{Re}^{-2}\text{Fr}^{-4}} + 1 \right)} \right), \end{aligned} \quad (5.30)$$

which does allow a wavy solution, but a solution which damps out very quickly in  $x$ -direction for  $\text{Re} \gg 1$ .

We proceed by doing analyses for the unsteady equations. The unsteady results to be obtained have a direct meaning for the steady equations as well; if no steady solution can be found for the unsteady formulation, then it also does not exist for the corresponding steady formulation either.

### 5.3 Analysis of semi-discrete problem

#### 5.3.1 For second-order accurate discretization of quasi free-surface condition

In this section we study the combined effect of the different truncation errors on the behaviour of the numerical approximation of the free-surface flow. For this purpose we consider the 2D, reduced Navier-Stokes equations from Section 5.2.2, discretize these in space according to (2.12) – (2.14), and next derive the corresponding system of modified equations. For a semi-discretization on a uniform Cartesian grid with mesh width  $h$  this system reads

$$\begin{aligned} u_t + Uu_x + \varphi_x - \text{Re}^{-1}u_{zz} &= U\frac{h^2}{3}u_{xxx} + \frac{h^3}{12}\varphi_{xxxx} + \text{Re}^{-1}\frac{h^2}{12}u_{zzzz} + \mathcal{O}(h^4), \\ w_t + Uw_x + \varphi_z - \text{Re}^{-1}w_{zz} &= U\frac{h^2}{3}w_{xxx} + \frac{h^3}{12}\varphi_{zzzz} + \text{Re}^{-1}\frac{h^2}{12}w_{zzzz} + \mathcal{O}(h^4), \\ u_x + w_z &= \frac{h^2}{3}u_{xxx} + \frac{h^3}{12}w_{zzzz} + \mathcal{O}(h^4), \end{aligned} \quad (5.31)$$

as  $h \rightarrow 0$ . On the free-surface boundary  $z = H(x, t)$ , the modified quasi free-surface condition

$$\varphi_t + U\varphi_x - \text{Fr}^{-2}w = U\frac{h^2}{3}\varphi_{xxx} + \mathcal{O}(h^3), \quad \text{as } h \rightarrow 0, \quad (5.32)$$

is imposed. For  $z \downarrow -\infty$  we assume that the uniform horizontal flow is recovered, and periodicity is assumed in  $x$ -direction.

We proceed by applying the same Fourier approach as in Section 5.2. Substitution of an isolated wave solution (5.9) into the system of modified equations (5.31), neglecting the higher than  $\mathcal{O}(h^2)$  terms, results in the following Fourier symbol

$$\hat{\mathbf{L}}_h(k, s\omega) := \begin{pmatrix} \hat{H}_h(k, s, \omega) & 0 & \hat{G}_{x,h} \\ 0 & \hat{H}_h(k, s, \omega) & \hat{G}_{z,h} \\ \hat{D}_{x,h} & \hat{D}_{z,h} & 0 \end{pmatrix}, \quad (5.33)$$

with

$$\hat{H}_h(k, s, \omega) := -i\omega + Uik \left(1 + \frac{h^2k^2}{3}\right) - \text{Re}^{-1}s^2 \left(1 + \frac{h^2s^2}{12}\right), \quad (5.34)$$

representing the Fourier symbol of the discrete convection-diffusion operator. The Fourier symbol of the discrete pressure gradient is given by

$$\hat{\mathbf{G}}_h = (ik, s)^T, \quad (5.35)$$

and

$$\hat{\mathbf{D}}_h = ik \left( 1 + \frac{h^2 k^2}{3}, s \right)^T \quad (5.36)$$

represents the discrete Fourier symbol of the divergence operator. From a straightforward analysis it follows that if the condition, originating from condition (5.14b),

$$k^2 \left( 1 + \frac{h^2 k^2}{3} \right) - s^2 = 0 \quad (5.37)$$

is satisfied, then

$$\text{span} \left\{ \begin{pmatrix} \hat{G}_{h,x} \\ \hat{G}_{h,z} \\ -\hat{H}_h(k, s(k), \omega) \end{pmatrix} \right\} \quad (5.38)$$

constitutes a basis of  $\mathcal{N}(\hat{\mathbf{L}}_h(k, s(k), \omega))$ . From (5.37), it follows that by good approximation for  $h \downarrow 0$ :

$$s(k) = \pm k \left( 1 + \frac{h^2 k^2}{6} \right). \quad (5.39)$$

The corresponding solution of the system of modified equations is

$$\mathbf{q}_h(x, z, t) = \begin{pmatrix} \hat{G}_{h,x} \\ \hat{G}_{h,z} \\ -\hat{H}_h(k, s(k), \omega) \end{pmatrix} e^{ikx + k \left( 1 + \frac{h^2 k^2}{6} \right) z - i\omega t}, \quad (5.40)$$

where we have used the positive root of (5.39) due to the fact that for  $z \downarrow -\infty$  the solution perturbations are zero.

The dispersion relation for the modified problem follows by substitution of this solution into (5.32) yielding

$$i\omega \hat{H}_h - U i \tilde{k} \hat{H}_h - \text{Fr}^{-2} \hat{G}_{h,z} = 0, \quad (5.41)$$

which has the roots

$$\omega_{1,2} = U \tilde{k} \pm \frac{1}{2} \sqrt{4\text{Fr}^{-2} s - \text{Re}^{-2} \tilde{s}^4} + i \frac{1}{2} \text{Re}^{-1} \tilde{s}^2, \quad (5.42)$$

where we have introduced the following short-hand notations

$$\tilde{k} := k \left( 1 + \frac{h^2 k^2}{3} \right), \quad \tilde{s}^2 := s^2 \left( 1 + \frac{h^2 s^2}{12} \right), \quad (5.43)$$

and in which  $s = k \left(1 + \frac{h^2 k^2}{6}\right)$ . Note that, as for the continuous equations, due to the presence of the positive imaginary  $\text{Re}^{-1}$ -term, according to this Fourier analysis, here steady waves cannot exist either. (And, again, the sign of the term under the square root cannot change this.)

In the inviscid limit, (5.42) reduces to

$$\omega_{1,2} = Uk \left(1 + \frac{h^2 k^2}{3}\right) \pm \sqrt{\text{Fr}^{-2} k \left(1 + \frac{h^2 k^2}{6}\right)}. \quad (5.44)$$

A stationary wave exists if

$$U^2 k \left(1 + \frac{h^2 k^2}{3}\right) = \text{Fr}^{-2} \left(1 + \frac{h^2 k^2}{6}\right). \quad (5.45)$$

Expansion of this result for  $h \rightarrow 0$  yields

$$U^2 k \left(1 + \frac{h^2 k^2}{2}\right) = \text{Fr}^{-2}. \quad (5.46)$$

This relation implies that for fixed  $U$  and  $\text{Fr}$ ,  $k$  decreases with increasing  $h$ . I.e., the coarser the mesh, the larger the length of the free-surface wave. Using the fact that for  $h = 0$  the solution  $k$  of (5.46) reads  $k = \frac{1}{(U\text{Fr})^2}$ , for small  $h$  we take as the solution:

$$k = \frac{1}{(U\text{Fr})^2} (1 + \epsilon), \quad |\epsilon| \ll 1. \quad (5.47)$$

Substitution of (5.47) into (5.46) yields, neglecting higher-order terms,  $\epsilon = \frac{-\frac{1}{2}h^2}{(U\text{Fr})^4}$ , i.e.,

$$k = \frac{1}{(U\text{Fr})^2} \left(1 - \frac{\frac{1}{2}h^2}{(U\text{Fr})^4}\right). \quad (5.48)$$

We investigate the correctness of (5.48) for the experimental results depicted in Figure 2.13. To obtain these results we have applied three grids with in  $x$ -direction the mesh sizes  $\frac{h}{2}$ ,  $h$  and  $2h$ , with  $h = \frac{6}{80}$ . For  $U$  and  $\text{Fr}$  we have taken  $U = 1$  and  $\text{Fr} = 0.6$ . Through the wavelength relation  $\lambda = \frac{2\pi}{k}$ , according to (5.48) it should then hold:

$$\lambda_{\frac{h}{2}} = 2.27, \quad \lambda_h = 2.31, \quad \lambda_{2h} = 2.48, \quad (5.49)$$

with  $\lambda_{h=0} = 2.26$ . Detailed observation of the results depicted in Figure 2.13 shows that the above, analytically found wavelengths are quite accurate, thus proving that (5.48) is not only of *qualitative* value but even of *quantitative* value. This is the promised, more rigorous analytical explanation for the behaviour of the dispersion error observed in Figure 2.13.

Note that the above error term does not solely come from the truncation error of the quasi free-surface condition. This can be verified by redoing the analysis for the exact reduced Navier-Stokes equations and the modified quasi free-surface condition (5.32). Then, in Fourier form, the quasi free-surface condition reads

$$i\omega \hat{H} - U i k \hat{H} - \text{Fr}^{-2} \hat{G}_z = 0, \quad (5.50)$$



with  $\hat{H} = -i\omega + Uik - \text{Re}^{-1}k^2$ ,  $\hat{G}_z = k$  and  $\tilde{k} = k \left(1 + \frac{h^2k^2}{3}\right)$ . In the inviscid limit this yields

$$\omega_{1,2} = Uk \left(1 + \frac{h^2k^2}{6}\right) \pm \sqrt{(Uk)^2 \left(\frac{h^2k^2}{6}\right)^2 + \text{Fr}^{-2}k}. \quad (5.51)$$

Hence, for steady waves it holds

$$U^2k \left(1 + \frac{h^2k^2}{3}\right) = \text{Fr}^{-2}, \quad (5.52)$$

which differs from (5.46), but which agrees – as expected – with the early result (2.26) as far as it concerns the error term.

### 5.3.2 For first-order upwind discretization of quasi free-surface condition

In Section 4.3 we considered the results belonging to the  $\mathcal{O}(h)$ -upwind discretization of the quasi free-surface condition. The effect of this discretization on the quasi free-surface condition is analyzed in this section. The modified equation belonging to the  $\mathcal{O}(h)$ -upwind discretization is

$$\varphi_t + U\varphi_x - \text{Fr}^{-2}w = U\frac{h}{2}\varphi_{xx} + \mathcal{O}(h^2), \quad \text{as } h \rightarrow 0. \quad (5.53)$$

Applying the Fourier-transform to the previous expression and substitution of the solution (5.40) into the Fourier-transformed quasi free-surface condition yields the following expression for the dispersion relation

$$i\omega\hat{H}_h - \left(ik + \frac{hk^2}{2}\right)U\hat{H}_h - \text{Fr}^{-2}\hat{G}_{h,z} = 0, \quad (5.54)$$

where  $\hat{H}_h$  is according to (5.34) and  $\hat{G}_{h,z} = s$ . Equation (5.54) leads to a quadratic equation for  $\omega$  which, neglecting the  $\mathcal{O}(h^2)$  term, has the following zeros:

$$\omega_{1,2} = \frac{2Uk + i(\text{Re}^{-1}s^2 - \frac{h}{2}Uk^2) \pm \sqrt{4\text{Fr}^{-2}s - (\text{Re}^{-1}s^2 + \frac{h}{2}Uk^2)^2}}{2}. \quad (5.55)$$

Note that now, as real part of the solution, we have (at least)  $e^{(\text{Re}^{-1} - \frac{h}{2}U)k^2t}$ , where we have again neglected the  $\mathcal{O}(h^2)$  terms. Whereas for both the exact equations considered in Section 5.2 and the  $\mathcal{O}(h^2)$  modified equations considered in Section 5.3, the time behaviour is always exponentially growing, with the  $\mathcal{O}(h)$  upwind discretization of the quasi free-surface condition, it can be exponentially decreasing, i.e., it can lead to a steady wave. A steady wave is obtained if  $\text{Re}^{-1} - \frac{h}{2}U < 0$ , i.e., if

$$U\text{Re}\frac{h}{2} > 1. \quad (5.56)$$

Considering the case  $U = 1$ ,  $\ell = 1$ , where  $\ell$  is the reference length, this means that for mesh Reynolds numbers  $\text{Re}h > 2$  we may obtain steady waves due to the

numerical diffusion introduced by the  $\mathcal{O}(h)$  upwind discretization of the pressure derivative in the quasi free-surface condition. In practical computations, with  $h \div \text{Re}^{-\frac{1}{2}}$ , condition (5.56) is easily satisfied. It has also been satisfied in all our computations in Chapter 4 with the  $\mathcal{O}(h)$  upwind discretization of the quasi free-surface condition. The current analytical result might partly explain why for the Series 60 hull computations we have obtained convergence and divergence with the  $\mathcal{O}(h)$  and  $\mathcal{O}(h^2)$  discretization of the quasi free-surface condition, respectively.

In the inviscid limit, (5.55) reduces to

$$\omega_{1,2} = \frac{2Uk - i\frac{h}{2}Uk^2 \pm \sqrt{4\text{Fr}^{-2}s - \frac{h^2}{4}U^2k^4}}{2}, \quad (5.57)$$

from which it follows that a steady wave satisfies the dispersion relation

$$\left(1 + \frac{h^2k^2}{16}\right)U^2k = \text{Fr}^{-2} \left(1 + \frac{h^2k^2}{6}\right), \quad (5.58)$$

which can be rewritten by good approximation as

$$U^2k \left(1 - \frac{5}{48}h^2k^2\right) = \text{Fr}^{-2}. \quad (5.59)$$

First note that despite the  $\mathcal{O}(h)$  discretization of the quasi free-surface condition, the error in  $k$  for steady waves is  $\mathcal{O}(h^2)$ . The  $\mathcal{O}(h)$  error appears to influence the time accuracy only. In the same way as in Section 5.3.2, as the approximate solution of (5.59) we find

$$k = \frac{1}{(U\text{Fr})^2} \left(1 + \frac{5}{48} \frac{h^2}{(U\text{Fr})^4}\right). \quad (5.60)$$

Note that the discretization error in (5.60) is even smaller than in (5.48), and that as opposed to (5.48), (5.60) predicts a *shortening* of the wavelengths when coarsening the mesh.

## 5.4 Conclusions

For the exact, 2D equations we have found the remarkable result that when imposing the quasi free-surface boundary condition (and when linearizing, considering periodicity in  $x$ -direction, infinitely deep water, and a Fourier form of the solution), steady free-surface waves can exist for the full Navier-Stokes equations and their inviscid limit (the Euler equations), but *not* for the in-between equations which the reduced Navier-Stokes equations are. Steady free-surface waves cannot exist either for the  $\mathcal{O}(h^2)$  discretized reduced Navier-Stokes equations in combination with the  $\mathcal{O}(h^2)$  discretization of the quasi free-surface condition. However, for the  $\mathcal{O}(h^2)$  discretized reduced Navier-Stokes equations in combination with the  $\mathcal{O}(h)$  upwind discretization of the quasi free-surface condition, steady waves *can* exist if the mesh Reynolds number is larger than 2. It is noteworthy that the dispersion error for these latter waves is not yet of first order but still of second order.

---

The analysis performed in this chapter explains why in Chapter 4 we could obtain steady free-surface waves with the  $\mathcal{O}(h)$ -upwind discretized quasi free-surface condition and why we could *not* obtain these waves with the  $\mathcal{O}(h^2)$ -upwind discretized quasi free-surface condition. (Given the negligible influence of viscous phenomena in Chapter 2, there the  $\mathcal{O}(h^2)$  discretization of the quasi free-surface condition *did* yield steady free-surface waves.) Computations with the *full* Navier-Stokes equations and the  $\mathcal{O}(h^2)$ -upwind discretization of the quasi-free surface condition are still hampered by mesh-tangling problems.



## *Part B: Capturing Method*



## CHAPTER 6

# Linearized Godunov Scheme for Two-Fluid Flow Computations

### 6.1 Introduction

This section will serve as a brief introduction to the second part of this thesis, consisting of the Chapters 6, 7 and 8. As pointed out in the introduction of this thesis; free-surface methods can be roughly divided into two classes: *moving-grid* techniques and *fixed-grid* techniques. The former have been considered in Chapters 2–5. The latter will be the subject of Chapters 6, 7 and 8. A general exposition of capturing methods has already been given in Chapter 1. Here, we will confine ourselves to a small resume.

In capturing methods, the free surface is determined through solving the initial-boundary-value problem governing the two-fluid flow problem, *without* imposing free-surface conditions. The location of the interface can be elegantly defined with the level-set method. A recent overview of this method is provided in [54]. This method is also adopted in these chapters, where the interface is defined as the ‘zero level-set’. Here we will employ a finite-volume discretization of the governing partial differential equations and boundary conditions. This discretization method requires an evaluation of the flux functions at the cell faces. Our flux formulas follow from the (approximate) solution of a two-fluid Riemann problem, which is the subject of the present chapter. The next chapter will provide an analysis of the so-called ‘pressure-oscillation’ phenomenon, i.e., the zeroth-order accurate, non-monotone behaviour of the pressure across the contact discontinuity, which represents the two-fluid interface. A number of fixes for the pressure-oscillation problem, which have appeared in the literature, is investigated. From this investigation the ‘ghost-fluid’ method emerges as the best. This method will then be used in a flow problem where the gravity force is the sole external force driving the flow. This flow problem is investigated in Chapter 8.

In the present chapter, the emphasis lies on the development of a liquid-gas Godunov-type scheme, in combination with a level-set technique, i.e., on flows with a single spatial dimension only. The contents of the chapter are the following. In Section 6.2, the continuous flow model is given (conservation laws, equation of state

and level-set equation). In Section 6.3, the space discretization of the equations is presented (the Riemann problem and the corresponding Godunov-type scheme, at both interior and boundary cell faces). In Section 6.4, we show that this scheme is a linearized, two-fluid Godunov scheme.

## 6.2 Flow model

### 6.2.1 Conservation equations

We assume that the two fluids are immiscible. Denoting the densities of the two fluids by  $\rho_1$  and  $\rho_2$ , this implies that if  $\rho_1 > 0$  then  $\rho_2 = 0$  and – vice versa – if  $\rho_2 > 0$  then  $\rho_1 = 0$ . See Figure 6.1 for a 1D illustration with  $x_{fs}$  the location of the free surface.

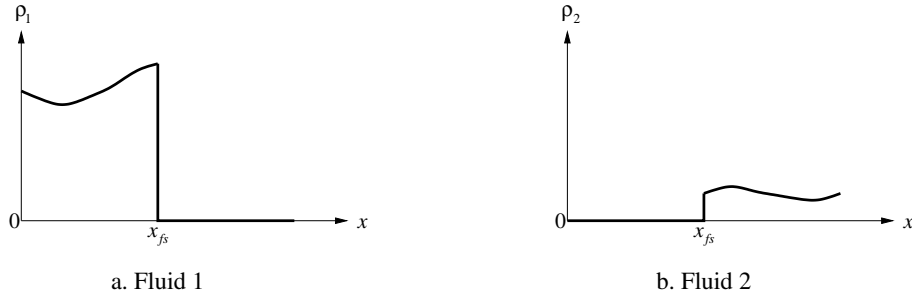


Figure 6.1: Density distributions in 1D flow of two immiscible fluids.

The immiscibility also implies that  $u_1(x_{fs}) = u_2(x_{fs}) = u(x_{fs})$ , i.e., the free surface is a contact discontinuity. The separate masses of the two fluids need to be conserved. In 1D, this means that for a stationary control volume  $\Omega$ :

$$\int_{\Omega} \frac{d\rho_1}{dt} dx + (\rho_1 u)_{\partial\Omega_{right}} - (\rho_1 u)_{\partial\Omega_{left}} = 0, \quad (6.1a)$$

$$\int_{\Omega} \frac{d\rho_2}{dt} dx + (\rho_2 u)_{\partial\Omega_{right}} - (\rho_2 u)_{\partial\Omega_{left}} = 0. \quad (6.1b)$$

For the control volume  $\Omega$  we may next define the bulk density as  $\rho \equiv \frac{\int_{\Omega} (\rho_1 + \rho_2) dx}{\int_{\Omega} dx}$ , which, for  $\rho_i \in C^0[V_i]$  ( $i = 1, 2$ ), is equal to  $\rho = \frac{\rho_1 V_1 + \rho_2 V_2}{V_1 + V_2}$ , where  $V_1$  and  $V_2$  are the sizes of the subvolumes of  $\Omega$  that are filled with fluid 1 and fluid 2, respectively. Introducing the volume fraction  $\alpha$  of fluid 1,  $\alpha = \frac{V_1}{V_1 + V_2}$ , we can write

$$\rho = \alpha \rho_1 + (1 - \alpha) \rho_2, \quad \alpha \in [0, 1]. \quad (6.2)$$

An alternative for (6.1) is then

$$\int_{\Omega} \frac{d\rho}{dt} dx + (\rho u)_{\partial\Omega_{right}} - (\rho u)_{\partial\Omega_{left}} = 0, \quad (6.3)$$



plus a still to be determined equation for the location(s)  $x_{fs}$  of the interface(s) in space and time. The latter equation determines  $\alpha = \alpha(x, t)$ . Looking ahead at the differences between both formulations in a numerical implementation, we already note that with the bulk-density formulation, in a finite-volume discretization, total mass of the fluid will be conserved, but not necessarily the masses of the two separate fluids. In case  $\alpha(x, t)$  is not exactly resolved, the two separate masses are not conserved. As opposed to that, with formulation (6.1) in a finite-volume discretization, the masses of the separate fluids are always exactly conserved. Hence, when using formulation (6.3), an accurate resolution of the interface location(s) is of paramount importance.

As far as momentum is concerned, the bulk density is a more practical quantity than the densities of the two separate fluids, because – as opposed to mass – the momenta of the two fluids do not need to be conserved separately. Only the total amount does, the two fluids can exchange momentum. Because surface tension is not considered, it also holds  $p_1(x_{fs}) = p_2(x_{fs}) = p(x_{fs})$ . So, for the momentum equation we can write

$$\int_{\Omega} \frac{d(\rho u)}{dt} dx + (\rho u^2 + p)_{\partial\Omega_{right}} - (\rho u^2 + p)_{\partial\Omega_{left}} = 0. \quad (6.4)$$

To describe two-fluid flows, we opt for the bulk-density description, i.e., equations (6.2)–(6.4). This system of equations is not yet balanced. There are six unknowns:  $\rho_1, \rho_2, \rho, p, u$  and  $\alpha$ . The equation for the location of the interface (determining  $\alpha$ ) still has to be chosen. For this, we follow a level-set approach, to be discussed in the next section. For the remaining unknowns  $\rho_1$  and  $\rho_2$ , equations of state  $\rho_1(p)$  and  $\rho_2(p)$  are chosen in Section 6.2.3.

### 6.2.2 Level-set equation

As mentioned above, to conserve the masses of the separate fluids as accurately as possible when using the bulk-density formulation, it is essential to resolve the free-surface locations as accurately as possible. For that purpose, the level-set approach is (in principle) better suited than the VOF approach, because of its better smoothness properties. Good smoothness of the level-set function is first taken care of in the level-set function's initialization. A common approach is to initialize the level-set function as the signed distance to the initial free surface, with the distance positive in – say – fluid 1 and negative in fluid 2. The choice for the signed-distance function may not be attractive though. In case of, e.g., a 1D flow problem with two interfaces (Figure 6.2), the level-set function initialized as the signed-distance function would look as depicted in Figure 6.3a, i.e., perfectly smooth at both free surfaces, but with a non-differentiability in between. Denoting the level-set function by  $\phi$ , in formula, the function in Figure 6.3a reads

$$\phi(x) = \min(x - (x_{fs})_1, (x_{fs})_2 - x). \quad (6.5a)$$

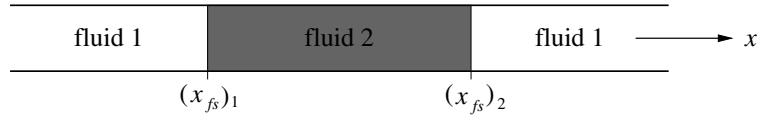


Figure 6.2: 1D two-fluid flow with two interfaces.

The initial level-set function does not need to be the signed-distance function. An alternative for it, which is uniformly smooth, is sketched in Figure 6.3b. In formula, here, one may think of, e.g.,

$$\phi(x) = e^{-(x-(x_{fs})_1)^2(x-(x_{fs})_2)^2} - 1. \quad (6.5b)$$

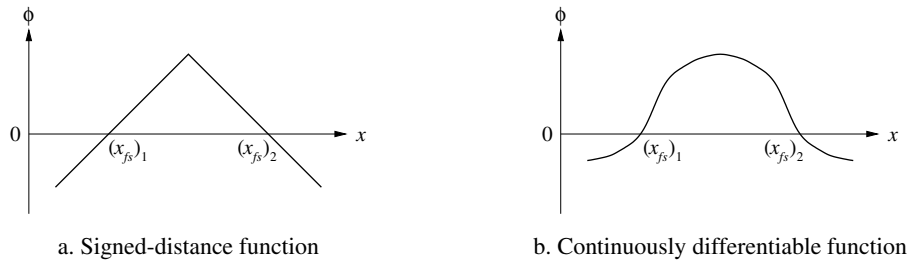


Figure 6.3: Possible initial level-set functions for 1D two-fluid flow with two interfaces.

In [44], numerical experiments are performed with level-set functions similar to the latter. For our present applications, the level-set function does not need to be uniformly smooth, except at the interfaces. Therefore, in the present chapter we adopt the signed-distance function as the initial level-set function.

In 1D, the level-set function is advected by

$$\frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} = 0, \quad (6.6)$$

with  $u$  the local velocity. Combined with bulk-mass conservation equation (6.3), quasi-linear equation (6.6) may be written in the conservative form

$$\int_{\Omega} \frac{d(\rho\phi)}{dt} dx + (\rho u \phi)_{\partial\Omega_{right}} - (\rho u \phi)_{\partial\Omega_{left}} = 0. \quad (6.7)$$

However, because  $\rho\phi$  is not a physical quantity, there is no need to conserve it. The form (6.7) is simply practical because it is consistent with the system (6.3)–(6.4), it can be directly added to it.

As soon as the level-set function becomes insufficiently smooth during its advection, assuming there is a smoothness criterion available, it must be regularized. Crucial in this reinitialization is that the free-surface location(s) at that specific moment are preserved as accurately as possible. If this is not taken care of, the reinitialization can be even counterproductive.

The advantages of level-set methods over particularly VOF methods are:

- Level-set functions are smooth at physical discontinuities and – hence – can be advected in a numerically accurate way precisely there. As opposed to that, the less smooth VOF function may easily smear out or become non-monotonous during its advection, thus deteriorating the resolution of the free surface.
- The level-set equation can be directly embedded into the system of fluid-flow equations and discretized collectively and consistently with these. E.g., it can be included into the Godunov-type scheme, which is what we will do. Related to this, there is no principal difficulty in extending a 1D level-set technique to multi-D.

### 6.2.3 Equations of state

In our water-air computations, for both fluids, elegant use can be made of a single equation of state, Tait's [9, 71]:

$$\frac{p + Bp_\infty}{(1 + B)p_\infty} = \left( \frac{\rho}{\rho_\infty} \right)^\gamma, \quad (6.8)$$

where the subscript  $\infty$  refers to some reference state. For water, it holds (at sea level conditions):  $\gamma = 7$ ,  $B = 3000$ ,  $\rho_\infty = 1000 \text{ kg/m}^3$ , and for air:  $\gamma = \frac{7}{5}$ ,  $B = 0$ ,  $\rho_\infty = 1 \text{ kg/m}^3$ . With (6.8), both the water and air density, to be denoted from now on by  $\rho_w(p)$  and  $\rho_a(p)$ , are convex functions of pressure. Likewise, the corresponding bulk density is:

$$\rho(\phi, p) = \alpha(\phi)\rho_w(p) + (1 - \alpha(\phi))\rho_a(p), \quad \alpha(\phi) \in [0, 1]. \quad (6.9)$$

The physical consequences of this overall convexity are that neither locally very low speeds of sound (much lower than in pure water or pure air), nor entropy-condition-satisfying expansion shocks can occur. These two anomalous phenomena are typical for flows with non-convex equations of state, flows with, e.g., condensation or vaporization [29], and cannot occur in the immiscible two-fluid flows considered here. To give some more evidence of this, consider the speed of sound of the bulk fluid:  $c^2 = \frac{\partial p}{\partial \rho}$ . Using (6.9), we can write  $\frac{1}{c^2} = \frac{\alpha}{c_w^2} + \frac{1-\alpha}{c_a^2}$ , with  $c_w^2 = \left( \frac{d\rho_w}{dp} \right)^{-1}$  and  $c_a^2 = \left( \frac{d\rho_a}{dp} \right)^{-1}$ . So,

$$c^2 = \frac{c_w^2 c_a^2}{\alpha c_a^2 + (1 - \alpha) c_w^2}. \quad (6.10)$$

Assuming that for any given  $p$ ,  $c_a < c_w$ , from (6.10) it is seen that  $c_a \leq c \leq c_w$  for all  $\alpha \in [0, 1]$ . In two-phase flows with condensation or vaporization, the pressure-density diagram may look as sketched in Figure 6.4a, i.e., as a non-convex curve with extremely small values of the speed of sound,  $\frac{dp}{d\rho}$ , in the condensation/vaporization zone. As opposed to that, in the case of two immiscible fluids, a family of purely convex curves exists, curves that become increasingly steeper for increasing  $\alpha$  (Figure 6.4b). So, for any  $p$  and for all values of  $\alpha \in (0, 1)$  it holds  $c_a < c < c_w$ .

A slight inconvenience of the nonlinear equation of state (6.8) in combination with (6.9) is that the calculation of  $p$  for known  $\rho$  and  $\alpha$ ,  $\alpha \in (0, 1)$ , needs to be done numerically.

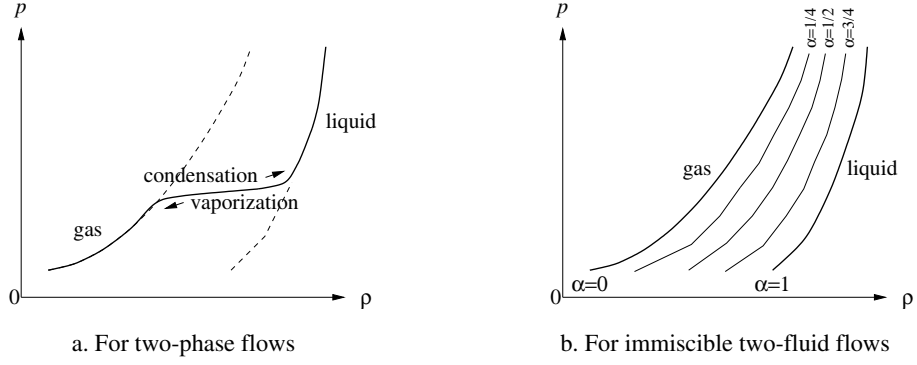


Figure 6.4: Pressure-density diagrams.

## 6.3 Discretization

### 6.3.1 Finite volumes

Summarizing, for a control volume  $\Omega$ , the system of equations considered reads

$$\int_{\Omega} \frac{dq}{dt} dx + (f(q))_{\partial\Omega_{right}} - (f(q))_{\partial\Omega_{left}} = 0, \quad (6.11a)$$

with the state vector  $q : \Omega \times \mathbb{R}^+ \rightarrow \mathbb{R}^3$  and the flux function  $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  given by

$$q = \begin{pmatrix} \rho \\ \rho u \\ \rho \phi \end{pmatrix}, \quad f(q) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u \phi \end{pmatrix}, \quad (6.11b)$$

where

$$\rho = \alpha(\phi)\rho_w(p) + (1 - \alpha(\phi))\rho_a(p), \quad (6.11c)$$

and

$$\frac{\rho_w(p)}{(\rho_w)_{\infty}} = \left( \frac{p + B_w p_{\infty}}{(1 + B_w)p_{\infty}} \right)^{\frac{1}{\gamma_w}}, \quad \frac{\rho_a(p)}{(\rho_a)_{\infty}} = \left( \frac{p + B_a p_{\infty}}{(1 + B_a)p_{\infty}} \right)^{\frac{1}{\gamma_a}}, \quad (6.11d)$$

and with  $\alpha(\phi)$  the fraction of the size of  $\Omega$  over which  $\phi \geq 0$ . The natural discretization for (6.11a) is a finite-volume technique. We consider cell-centered finite volumes with, for simplicity, constant mesh size  $h$ . This choice directly allows us to work out the discretization of  $\alpha(\phi)$ . Consider finite volume  $\Omega_i$  and its left and right

neighbours (Figure 6.5) and define the level-set values at the cell faces  $\partial\Omega_{i-\frac{1}{2}}$  and  $\partial\Omega_{i+\frac{1}{2}}$  as

$$\phi_{i-\frac{1}{2}} = \frac{1}{2}(\phi_{i-1} + \phi_i), \quad \phi_{i+\frac{1}{2}} = \frac{1}{2}(\phi_i + \phi_{i+1}). \quad (6.12)$$

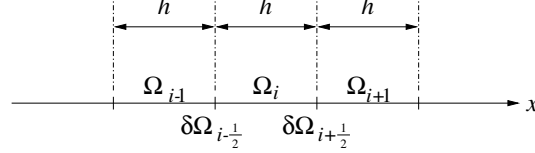


Figure 6.5: Finite volume  $\Omega_i$  and its two neighbours.

Then, for  $\phi_i \geq 0$ , we propose the following expression for  $\alpha_i$ :

$$\phi_{i-\frac{1}{2}} \geq 0, \phi_{i+\frac{1}{2}} \geq 0: \quad \alpha_i = 1, \quad (6.13a)$$

$$\phi_{i-\frac{1}{2}} < 0, \phi_{i+\frac{1}{2}} \geq 0: \quad \alpha_i = \frac{1}{2} \left( \frac{\phi_i}{\phi_i - \phi_{i-\frac{1}{2}}} + 1 \right), \quad (6.13b)$$

$$\phi_{i-\frac{1}{2}} \geq 0, \phi_{i+\frac{1}{2}} < 0: \quad \alpha_i = \frac{1}{2} \left( 1 + \frac{\phi_i}{\phi_i - \phi_{i+\frac{1}{2}}} \right), \quad (6.13c)$$

$$\phi_{i-\frac{1}{2}} < 0, \phi_{i+\frac{1}{2}} < 0: \quad \alpha_i = \frac{1}{2} \left( \frac{\phi_i}{\phi_i - \phi_{i-\frac{1}{2}}} + \frac{\phi_i}{\phi_i - \phi_{i+\frac{1}{2}}} \right). \quad (6.13d)$$

The four possibilities in (6.13) are illustrated in Figure 6.6. So, in determining  $\phi_{i-\frac{1}{2}}$  and  $\phi_{i+\frac{1}{2}}$ , as well as  $x(\phi = 0)$ , use is made of piecewise linear interpolation of  $\phi$ . The linear interpolation is exact as long as the level-set function is the signed-distance function. As soon as this distance property is lost, the exact conservation of the separate water and air masses is lost. For  $\phi_i < 0$ , similar expressions can be written. Using this similarity, the system of expressions for  $\alpha_i$  can be coded compactly.

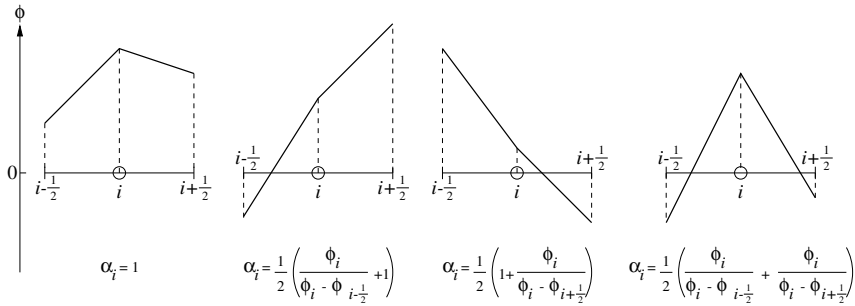


Figure 6.6: Four possible combinations of signs of  $\phi_{i-\frac{1}{2}}$  and  $\phi_{i+\frac{1}{2}}$ ,  $\phi_i \geq 0$ .

### 6.3.2 Riemann-problem approach

For the control-volume formulation, we need a formula for the flux vector across a cell face. The formula must have built-in physics for accurately capturing the free surface. For that purpose, trivial, term-by-term flux formulas such as, e.g.,  $(f(q))_{i+\frac{1}{2}} = f(\frac{1}{2}(q_i + q_{i+1}))$  are less appropriate than formulas derived from the Riemann problem. Besides a good capturing of the free surface, a Riemann problem approach is expected to yield robustness and a good boundary-condition treatment. For ship-hydrodynamics applications, a physically proper discretization of convection terms may be as relevant as for aircraft and spacecraft aerodynamics. The convection phenomena in ship hydrodynamics are less rich (no supersonic speeds), but the Reynolds numbers are generally higher (up to  $\mathcal{O}(10^9)$ ).

The exact solution of the 1D Riemann problem on each cell face, Godunov's approach [26], requires the exact computation of the cell-face state. For the current equations, this implies the use of a numerical root finder. We avoid this by considering an approximate Riemann solver: Osher's [56], to start with. Denoting the left and right cell-face state by  $q_0$  and  $q_1$ , and the flux formula by  $F(q_0, q_1)$ , Osher's approximate Riemann solver may be written as

$$F(q_0, q_1) = f(q_0) + \int_{q_0}^{q_1} \frac{df^-}{dq} dq, \quad (6.14)$$

with  $\frac{df^-}{dq}$  the negative eigenvalue part of the Jacobian  $\frac{df}{dq}$ . The eigenvalues of the present Jacobian are:  $\lambda_1 = u - \sqrt{\frac{\partial p}{\partial \rho}}$ ,  $\lambda_2 = u$ ,  $\lambda_3 = u + \sqrt{\frac{\partial p}{\partial \rho}}$ . Note that  $\frac{\partial p}{\partial \phi}$  does not occur in the wave speeds. The right eigenvectors are

$$\mathbf{r}_1 = \begin{pmatrix} 1 \\ u - \sqrt{\frac{\partial p}{\partial \rho}} \\ \phi \end{pmatrix}, \mathbf{r}_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \mathbf{r}_3 = \begin{pmatrix} 1 \\ u + \sqrt{\frac{\partial p}{\partial \rho}} \\ \phi \end{pmatrix}. \quad (6.15)$$

The Riemann-invariant relations describing the two intermediate states  $q_{\frac{1}{3}}$  and  $q_{\frac{2}{3}}$  along the wave path in state space, see Figure 6.7, are

$$u_{\frac{1}{3}} + \int^{\rho_{\frac{1}{3}}} \frac{1}{\rho} \sqrt{\frac{\partial p}{\partial \rho}} d\rho = u_0 + \int^{\rho_0} \frac{1}{\rho} \sqrt{\frac{\partial p}{\partial \rho}} d\rho, \quad (6.16a)$$

$$\phi_{\frac{1}{3}} = \phi_0, \quad (6.16b)$$

$$u_{\frac{1}{3}} = u_{\frac{2}{3}} = u_{\frac{1}{2}}, \quad (6.17a)$$

$$p_{\frac{1}{3}} = p_{\frac{2}{3}} = p_{\frac{1}{2}}, \quad (6.17b)$$

$$u_{\frac{2}{3}} - \int^{\rho_{\frac{2}{3}}} \frac{1}{\rho} \sqrt{\frac{\partial p}{\partial \rho}} d\rho = u_1 - \int^{\rho_1} \frac{1}{\rho} \sqrt{\frac{\partial p}{\partial \rho}} d\rho, \quad (6.18a)$$

$$\phi_{\frac{2}{3}} = \phi_1. \quad (6.18b)$$

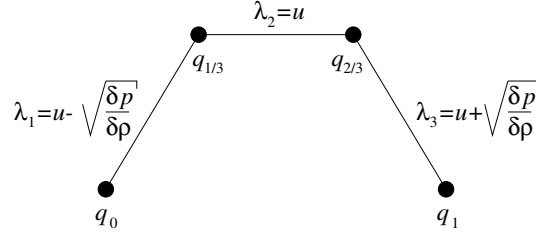


Figure 6.7: Wave path in state space for immiscible two-fluid flow.

The level-set function  $\phi$  can only change along the subpath corresponding with the eigenvalue  $\lambda_2$ , i.e., as it should be, across the contact discontinuity. It is invariant along the outer subpaths, i.e., along both subpaths the distance to the free surface is constant. Through the known Riemann invariants  $u$  and  $p$  at the contact discontinuity, the kinematic and dynamic free-surface boundary conditions (zero mass transport and stresses) are satisfied. The integrals in (6.16a) and (6.18a) can be written out explicitly for the equations given in (6.11d). However, when a free surface is captured, i.e., when  $\phi_{\frac{1}{3}}$  and  $\phi_{\frac{2}{3}}$  differ in sign, explicit calculation of  $u_{\frac{1}{2}}$  and  $p_{\frac{1}{2}}$  is hampered by nonlinearity, the aforementioned drawback of Osher's scheme. (A transcendental equation needs to be solved.) Of course,  $u_{\frac{1}{2}}$  and  $p_{\frac{1}{2}}$  can then be determined numerically. Given foreseen future applications in which  $u_{\frac{1}{2}}$  and  $p_{\frac{1}{2}}$  need to be differentiated with respect to  $q_0$  and  $q_1$ , we do not do so and propose another approximate Riemann solver in the next section.

### 6.3.3 Two-fluid, linearized Godunov scheme

Since  $\phi$  is constant along the two outer subpaths of the wave path in Figure 6.7, along both subpaths the bulk density can only vary due to pressure changes. Because the flows to be considered are uniformly low-subsonic, large density changes will not occur there and – consequently – the integrals  $\int^{\rho_{\frac{1}{3}}} \frac{1}{\rho} \sqrt{\frac{\partial p}{\partial \rho}} d\rho$  and  $\int^{\rho_{\frac{2}{3}}} \frac{1}{\rho} \sqrt{\frac{\partial p}{\partial \rho}} d\rho$  can be linearized by good approximation around  $\rho_0$  and  $\rho_1$ , respectively. Linearization of (6.16a) and (6.18a) yields

$$u_{\frac{1}{2}} = u_0 - (\rho_{\frac{1}{3}} - \rho_0) \frac{c_0}{\rho_0}, \quad (6.19a)$$

$$u_{\frac{1}{2}} = u_1 + (\rho_{\frac{2}{3}} - \rho_1) \frac{c_1}{\rho_1}. \quad (6.19b)$$

Likewise,  $p_{\frac{1}{2}}$  can be linearized around  $\rho_0$  and  $\rho_1$ :

$$p_{\frac{1}{2}} = p_0 + (\rho_{\frac{1}{3}} - \rho_0) c_0^2, \quad (6.20a)$$

$$p_{\frac{1}{2}} = p_1 + (\rho_{\frac{2}{3}} - \rho_1) c_1^2. \quad (6.20b)$$

Elimination of  $\rho_{\frac{1}{3}} - \rho_0$  and  $\rho_{\frac{2}{3}} - \rho_1$  from (6.19) and (6.20) yields

$$\frac{p_{\frac{1}{2}} - p_0}{u_{\frac{1}{2}} - u_0} = -C_0, \quad C_0 \equiv \rho_0 c_0, \quad (6.21a)$$

$$\frac{p_{\frac{1}{2}} - p_1}{u_{\frac{1}{2}} - u_1} = C_1, \quad C_1 \equiv \rho_1 c_1, \quad (6.21b)$$

i.e.,

$$\begin{pmatrix} u_{\frac{1}{2}} \\ p_{\frac{1}{2}} \end{pmatrix} = \begin{pmatrix} \frac{C_0 u_0 + C_1 u_1 + (p_0 - p_1)}{C_0 + C_1} \\ \frac{C_1 p_0 + C_0 p_1 + C_0 C_1 (u_0 - u_1)}{C_0 + C_1} \end{pmatrix}, \quad (6.22a)$$

with for the density and level-set function in the two intermediate points:

$$\begin{pmatrix} \rho_{\frac{1}{3}} \\ \phi_{\frac{1}{3}} \end{pmatrix} = \begin{pmatrix} \rho_0 + \frac{p_{\frac{1}{2}} - p_0}{c_0^2} \\ \phi_0 \end{pmatrix}, \quad (6.22b)$$

$$\begin{pmatrix} \rho_{\frac{2}{3}} \\ \phi_{\frac{2}{3}} \end{pmatrix} = \begin{pmatrix} \rho_1 + \frac{p_{\frac{1}{2}} - p_1}{c_1^2} \\ \phi_1 \end{pmatrix}. \quad (6.22c)$$

Excluding all supersonic possibilities from the matrix in Figure 6.8, which shows all possible combinations of eigenvalue signs along the wave path – note the consequent improvement in efficiency – the two-fluid, linearized scheme reads then:

$$F(q_0, q_1) = f(q_{\frac{1}{3}}) = \begin{pmatrix} \rho_{\frac{1}{3}} u_{\frac{1}{2}} \\ \rho_{\frac{1}{3}} u_{\frac{1}{2}}^2 + p_{\frac{1}{2}} \\ \rho_{\frac{1}{3}} u_{\frac{1}{2}} \phi_{\frac{1}{3}} \end{pmatrix}, \quad \text{if } u_{\frac{1}{2}} \geq 0, \quad (6.23a)$$

$$F(q_0, q_1) = f(q_{\frac{2}{3}}) = \begin{pmatrix} \rho_{\frac{2}{3}} u_{\frac{1}{2}} \\ \rho_{\frac{2}{3}} u_{\frac{1}{2}}^2 + p_{\frac{1}{2}} \\ \rho_{\frac{2}{3}} u_{\frac{1}{2}} \phi_{\frac{2}{3}} \end{pmatrix}, \quad \text{if } u_{\frac{1}{2}} \leq 0. \quad (6.23b)$$



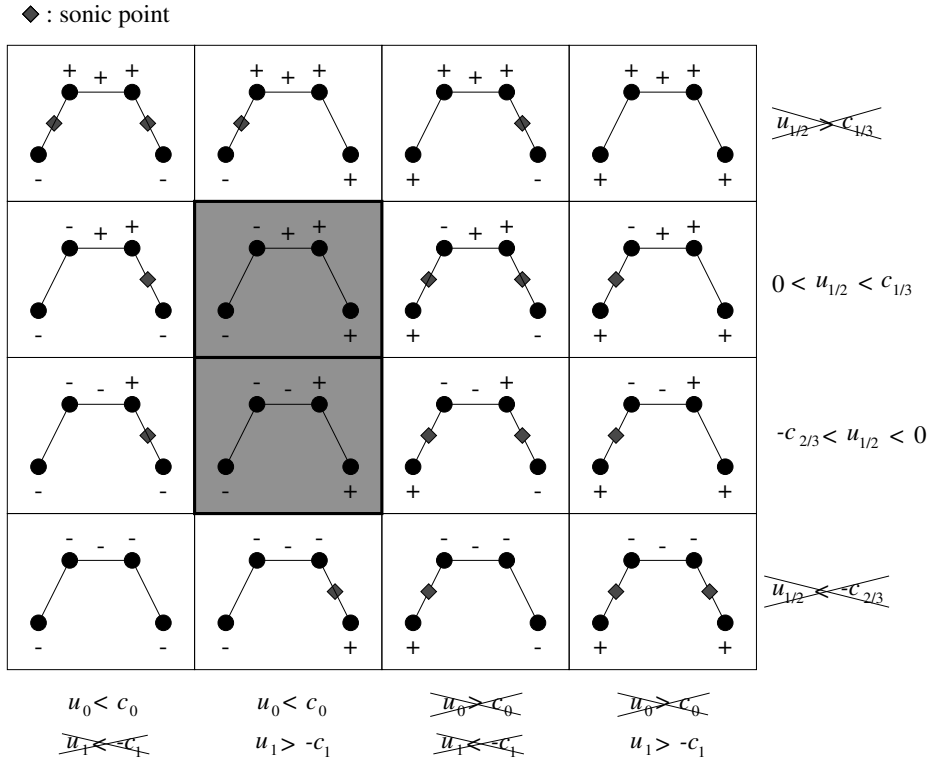


Figure 6.8: All possible eigenvalue-sign combinations along wave path from Figure 6.7, crossed out: all supersonic speeds.

Note that the real nonlinear flux functions  $f(q_{\frac{1}{3}})$  and  $f(q_{\frac{2}{3}})$  are applied, and not  $F(q_0, q_1) = f(q_0) + (q_{\frac{1}{3}} - q_0) \frac{df(q_0)}{dq}$  if  $u_{\frac{1}{2}} \geq 0$  and  $F(q_0, q_1) = f(q_1) + (q_{\frac{2}{3}} - q_1) \frac{df(q_1)}{dq}$  if  $u_{\frac{1}{2}} \leq 0$ . There is no need for the latter linearized formulas. On the contrary, as opposed to (6.22)–(6.23), they may give rise to an erroneous, ambiguous flux at  $u_{\frac{1}{2}} = 0$  (steady free surface);  $f(q_0) + (q_{\frac{1}{3}} - q_0) \frac{df(q_0)}{dq}$  and  $f(q_1) + (q_{\frac{2}{3}} - q_1) \frac{df(q_1)}{dq}$  may be different for  $u_{\frac{1}{2}} = 0$ .

For the single-fluid case, (6.23) reduces to

$$F(q_0, q_1) = \begin{pmatrix} \rho_{\frac{1}{2}} u_{\frac{1}{2}} \\ \rho_{\frac{1}{2}} u_{\frac{1}{2}}^2 + p_{\frac{1}{2}} \end{pmatrix}, \quad \rho_{\frac{1}{2}} = \rho(p_{\frac{1}{2}}), \quad (6.24)$$

with  $u_{\frac{1}{2}}$  and  $p_{\frac{1}{2}}$  still given by (6.22a). The latter scheme is known as the single-fluid, linearized Godunov scheme, see Section 2.3 and Appendix A in [46], and also Section 9.3 in [75]. In Section 6.4, through a (partial) derivation of the (exact) Godunov scheme, it is shown that the present two-fluid, linearized scheme (6.21)–(6.22) is in fact the two-fluid, linearized Godunov scheme. For clarity, in the following, we only use the latter name for scheme (6.22)–(6.23).

### 6.3.4 Boundary-condition treatment

The boundary of the computational domain is formed by cell faces. The fluxes across the boundary faces can be computed with (6.23) as well. Denoting the state at the boundary by  $q_b$ , in case of a left boundary  $q_0 = q_b$  and in case of a right boundary  $q_1 = q_b$ . For (subsonic) inflow, in 1D, two of the three components of  $q_b$  must be imposed, reducing the wave path of the two-fluid, linearized Godunov scheme, see Figure 6.9. In case of (subsonic) outflow, a single component of  $q_b$  must be imposed, leading to the reduced paths given in Figure 6.10.



Figure 6.9: Reduced wave paths for inflow boundary.

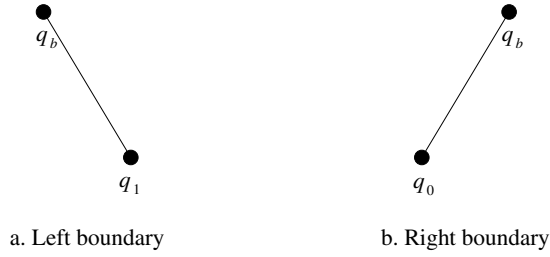


Figure 6.10: Reduced wave paths for outflow boundary.

With (6.14), for left inflow and left outflow it holds, as it should be,  $F(q_b, q_1) = f(q_b)$  and for both right cases, identically,  $F(q_0, q_b) = f(q_0) + f(q_b) - f(q_0) = f(q_b)$ . We work out the inflow and outflow boundary, and the non-permeable boundary as the limit case. For all three it holds, for a left boundary:

$$\frac{p_b - p_1}{u_b - u_1} = C_1, \quad (6.25a)$$

and at the right:

$$\frac{p_b - p_0}{u_b - u_0} = -C_0. \quad (6.25b)$$

#### Inflow boundary

The two boundary conditions to be imposed here cannot be  $u_b$  and  $p_b$  simultaneously; when  $u_b$  is imposed,  $p_b$  follows from (6.25) and vice versa. Hence, the second

boundary condition must be one for  $\phi_b$ . To compute the corresponding boundary flux  $f(q_b) = (\rho_b u_b, \rho_b u_b^2 + p_b, \rho_b \phi_b)^T$ , the '0D' bulk density still needs to be defined. In 2D and 3D, the bulk density can be computed in a normal 1D and 2D manner, respectively. In 1D, an appropriate '0D' choice is

$$\rho_b = \rho_w(\phi_b), \quad \text{for } \phi_b \geq 0, \quad (6.26a)$$

$$\rho_b = \rho_a(\phi_b), \quad \text{for } \phi_b < 0. \quad (6.26b)$$

### Outflow boundary

Here, in addition to (6.25a) and (6.25b), the equations

$$\phi_b = \phi_0, \quad (6.27a)$$

$$\phi_b = \phi_1 \quad (6.27b)$$

are available. So, the single boundary condition to be imposed must be  $u_b$ ,  $p_b$ , or some combination of both. The bulk density  $\rho_b$  is defined as in the inflow case.

### Non-permeable boundary

At a non-permeable boundary (at least)  $u_b = 0$  must be imposed, which, given (6.25), already determines  $p_b$ . Considering a non-permeable boundary as the limit case of an inflow boundary,  $\phi_b$  must still be imposed. Considering it as the limit of outflow,  $\phi_b$  follows from the interior solution ( $\phi_b = \phi_1$  for left boundary and  $\phi_b = \phi_0$  for right). The latter limit case is to be preferred for, e.g., ship-hydrodynamics applications. As opposed to the first limit case, it allows the water line to freely move up and down the ship hull, see Figure 6.11. Also here, the bulk density may be defined according to (6.26).

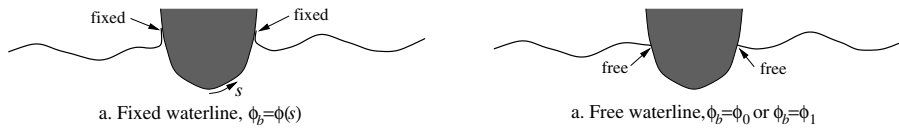


Figure 6.11: Cross section of ship hull with free surface.

## 6.4 The two-fluid Godunov scheme for Tait's equation of state

### 6.4.1 State space, Poisson curves and Hugoniot curves

For the present two-fluid case, just as for the single-fluid case, in physical space, the left and right Riemann states  $(u_0, p_0)$  and  $(u_1, p_1)$  are connected to the intermediate state  $(u_{1/2}, p_{1/2})$  through either a rarefaction wave or a shock wave. The four different possible pairs of shock and rarefaction waves that exist are sketched in Figure 6.12.

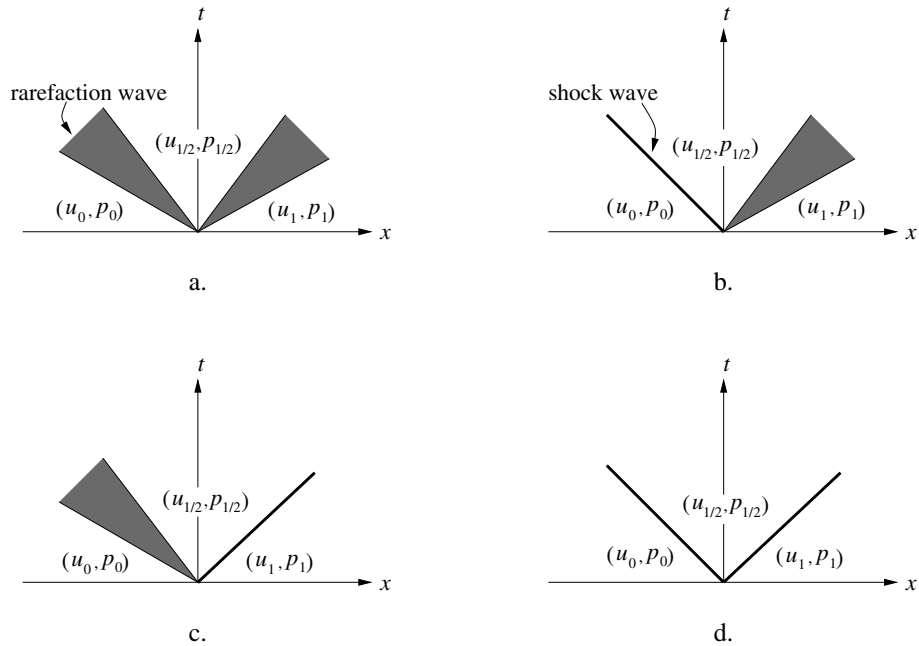


Figure 6.12: Wave combinations in physical space.

In the single-fluid case, the four states  $(u_{1/2}, p_{1/2})$  corresponding with those in Figure 6.12 are determined as the intersection in the  $(u, p)$ -state space of either two Poisson curves (Figure 6.13a), a Poisson and a Hugoniot curve (Figures 6.13b and 6.13c), or two Hugoniot curves (Figure 6.13d). The Poisson curve through a point  $(u_k, p_k)$ ,  $k = 0, 1$  represents all states  $(u, p)$  that can be reached from  $(u_k, p_k)$  through a single rarefaction wave, the Hugoniot curve all points reachable through a single shock wave. For details about this (single-fluid) theory, see Sections 80 and 81 in [21]. The specific forms of the Poisson and Hugoniot curves depend on the equation of state considered. In general, the curves are nonlinear. For a brief description of the two-fluid case on the basis of Tait's equation of state, see the next section.

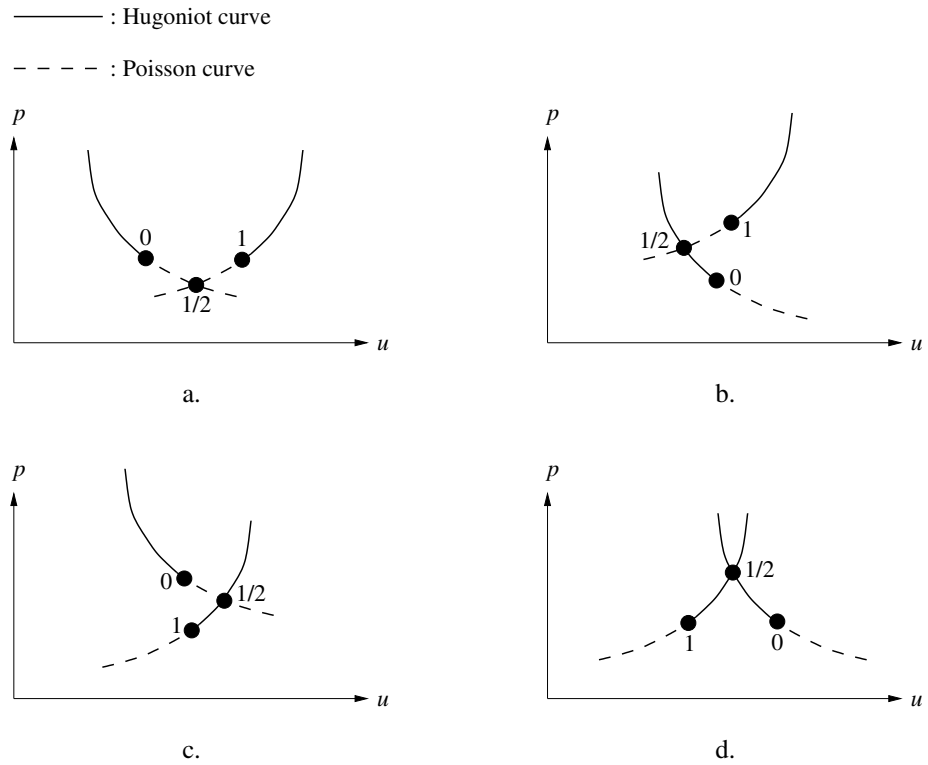


Figure 6.13: Poisson and Hugoniot curve combinations in state space, single-fluid, combinations a–d correspond with wave pairs in Figures 6.12a–d.

### 6.4.2 Families of Poisson curves

For a two-fluid model in which the bulk density description (6.11c) is used, through a point  $(u_k, p_k)$ ,  $k = 0, 1$  in state space, instead of a single Poisson curve, a family of Poisson curves exists, one curve for each value of the Volume-of-Fluid fraction  $\alpha_k$ ,  $\alpha_k \in [0, 1]$ ,  $k = 0, 1$ . For rarefaction waves connecting  $(u_0, p_0)$  to  $(u_{\frac{1}{2}}, p_{\frac{1}{2}})$ , the family is determined by

$$u + \int^p \frac{1}{\rho c} dp = u_0 + \int^{p_0} \frac{1}{\rho c} dp, \quad (6.28a)$$

with the bulk density  $\rho = \rho(\alpha_0, p)$  according to (6.11c) and the corresponding speed of sound  $c = c(\alpha_0, p)$  according to (6.10),  $\alpha_0 \in [0, 1]$ . Equation (6.28a) is equal to the first equation in (6.16a) with integration to  $p$  instead of to  $\rho$ . For rarefactions connecting  $(u_1, p_1)$  to  $(u_{\frac{1}{2}}, p_{\frac{1}{2}})$ , the family of Poisson curves is determined by

$$u - \int^p \frac{1}{\rho c} dp = u_1 - \int^{p_1} \frac{1}{\rho c} dp, \quad (6.28b)$$

with the similar expressions for  $\rho(\alpha_1, p)$  and  $c(\alpha_1, p)$ ,  $\alpha_1 \in [0, 1]$ . Finding formulas for the Poisson curves for general  $\alpha_0$  and  $\alpha_1$ ,  $(\alpha_0, \alpha_1) \in [0, 1] \times [0, 1]$ , is tedious. For convenience, here we consider the case  $\alpha_0 = \alpha_1 = 1$  only (100% water in both the left and right control volume). For  $\alpha = 1$ , with Tait's equation of state, it holds

$$\rho(p) = \left( \frac{p + B_w p_\infty}{(1 + B_w) p_\infty} \right)^{\frac{1}{\gamma_w}} (\rho_\infty)_w, \quad (6.29a)$$

$$c(p) = \left( \frac{p + B_w p_\infty}{(1 + B_w) p_\infty} \right)^{\frac{\gamma_w - 1}{2\gamma_w}} (c_\infty)_w, \quad (c_\infty)_w = \sqrt{\gamma_w \frac{(1 + B_w) p_\infty}{(\rho_\infty)_w}}. \quad (6.29b)$$

Substitution of (6.29a) and (6.29b) into (6.28a) and (6.28b), and integration, yields for the exact Poisson curves

$$u - u_0 = \frac{-2}{\gamma_w - 1} \left[ \left( \frac{p + B_w p_\infty}{(1 + B_w) p_\infty} \right)^{\frac{\gamma_w - 1}{2\gamma_w}} - \left( \frac{p_0 + B_w p_\infty}{(1 + B_w) p_\infty} \right)^{\frac{\gamma_w - 1}{2\gamma_w}} \right] (c_\infty)_w, \quad (6.30a)$$

for  $\alpha_0 = 1$ , and

$$u - u_1 = \frac{2}{\gamma_w - 1} \left[ \left( \frac{p + B_w p_\infty}{(1 + B_w) p_\infty} \right)^{\frac{\gamma_w - 1}{2\gamma_w}} - \left( \frac{p_1 + B_w p_\infty}{(1 + B_w) p_\infty} \right)^{\frac{\gamma_w - 1}{2\gamma_w}} \right] (c_\infty)_w, \quad (6.30b)$$

for  $\alpha_1 = 1$ . Linearization of (6.30a) and (6.30b) around  $(u_0, p_0)$  and  $(u_1, p_1)$ , respectively, gives, denoting  $\rho c$  by  $C$ :

$$\frac{p - p_0}{u - u_0} = \left( \frac{du}{dp} \right)_{p_0}^{-1} = -(C_0)_w, \quad \text{for } \alpha_0 = 1, \quad (6.31a)$$

$$\frac{p - p_1}{u - u_1} = \left( \frac{du}{dp} \right)_{p_1}^{-1} = (C_1)_w, \quad \text{for } \alpha_1 = 1, \quad (6.31b)$$

which is in agreement with the direct linearization of (6.16a) and (6.18a), which gives (6.21a) and (6.21b), respectively, which – in fact – already are the two families of linearized Poisson curves, with bulk densities  $\rho_0$  and  $\rho_1$  and speeds of sound  $c_0$  and  $c_1$  valid for all  $(\alpha_0, \alpha_1) \in [0, 1] \times [0, 1]$ .

### 6.4.3 Families of Hugoniot curves

In a Lagrangean formulation in which the shock wave is set still, across a shock connecting the pre-shock state  $(u_0, p_0)$  and the post-shock state  $(u_{\frac{1}{2}}, p_{\frac{1}{2}})$ , the flow is always from left to right, see Figure 6.14a. Vice versa, across a shock connecting the pre-shock state  $(u_1, p_1)$  and the post-shock state  $(u_{\frac{1}{2}}, p_{\frac{1}{2}})$ , the flow is always from right to left, see Figure 6.14b.

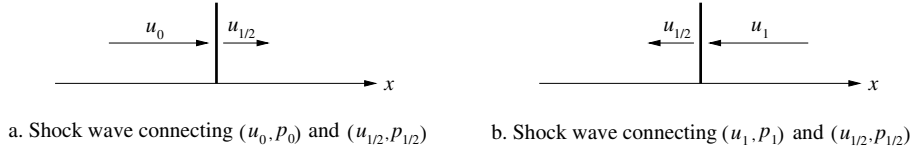


Figure 6.14: Shock waves in shock frame.

In the shock frame, the jump conditions corresponding with the situation of Figures 6.14a and 6.14b are, respectively

$$\begin{bmatrix} u \\ p \end{bmatrix} = m \begin{bmatrix} \frac{1}{\rho} \\ -u \end{bmatrix}, \quad (6.32a)$$

$$\begin{bmatrix} u \\ p \end{bmatrix} = m \begin{bmatrix} -\frac{1}{\rho} \\ u \end{bmatrix}, \quad (6.32b)$$

where  $m$  is the mass flow  $|\rho u|$  through the shock wave and  $[\cdot]$  denotes the jump operator. For a derivation of these shock relations in a Lagrangean formulation, see Section 62 in [21]. From both (6.32a) and (6.32b) it follows

$$m = \sqrt{\frac{[p]}{-[1/\rho]}}. \quad (6.33)$$

Using (6.33), the above two jump conditions for momentum can be written as, omitting the subscripts for the post-shock quantities,

$$p - p_0 = -\sqrt{\frac{p - p_0}{\frac{1}{\rho_0} - \frac{1}{\rho}}}(u - u_0), \quad (6.34a)$$

$$p - p_1 = \sqrt{\frac{p - p_1}{\frac{1}{\rho_1} - \frac{1}{\rho}}}(u - u_1). \quad (6.34b)$$

With in equation (6.34a) the bulk densities  $\rho_0 = \rho_0(\alpha_0, p_0)$  and  $\rho = \rho(\alpha_0, p)$  according to (6.11c), (6.34a) determines one family of Hugoniot curves, one curve for each value of  $\alpha_0 \in [0, 1]$ . Likewise, (6.34b) does for  $\alpha_1 \in [0, 1]$ . Equations (6.34a) and (6.34b) are the shock analogies of the rarefaction equations (6.28a) and (6.28b). Working out (6.34a) and (6.34b) for general  $\alpha_0$  and  $\alpha_1$  on the basis of Tait's equation of state is tedious. As far as the nonlinear equations are concerned, here we also restrict ourselves to the specific case  $\alpha_0 = \alpha_1 = 1$  (100% water in the left and right control volume). Elimination of the densities from (6.34a) and (6.34b) with Tait's equation of state yields after some rewriting

$$p - p_0 = -(C_0)_w \sqrt{\frac{1}{\gamma_w} \frac{\frac{p+B_w p_\infty}{p_0+B_w p_\infty} - 1}{1 - \left(\frac{p_0+B_w p_\infty}{p+B_w p_\infty}\right)^{\frac{1}{\gamma_w}}}}(u - u_0), \quad \text{for } \alpha_0 = 1, \quad (6.35a)$$

$$p - p_1 = (C_1)_w \sqrt{\frac{1}{\gamma_w} \frac{\frac{p+B_w p_\infty}{p_1+B_w p_\infty} - 1}{1 - \left(\frac{p_1+B_w p_\infty}{p+B_w p_\infty}\right)^{\frac{1}{\gamma_w}}}} (u - u_1), \quad \text{for } \alpha_1 = 1. \quad (6.35b)$$

Linearization of (6.35a) and (6.35b) around  $(u_0, p_0)$  and  $(u_1, p_1)$ , respectively, yields

$$\frac{p - p_0}{u - u_0} = \left(\frac{du}{dp}\right)_{p_0}^{-1} = -(C_0)_w, \quad \text{for } \alpha_0 = 1, \quad (6.36a)$$

$$\frac{p - p_1}{u - u_1} = \left(\frac{du}{dp}\right)_{p_1}^{-1} = (C_1)_w, \quad \text{for } \alpha_1 = 1, \quad (6.36b)$$

which is identical to (6.31a) and (6.31b) (as it should be because the Hugoniot curve and the Poisson curve through a point in state space are tangent to each other in that point).

Equations (6.34a) and (6.34b), which are valid for general  $\alpha_0$  and  $\alpha_1$ , respectively, can also be linearized more directly. For that purpose, rewrite both equations as

$$p - p_0 = -\sqrt{\rho\rho_0} \sqrt{\frac{p - p_0}{\rho - \rho_0}} (u - u_0), \quad \rho_0 = \rho_0(\alpha_0, p_0), \quad \rho = \rho(\alpha_0, p), \quad \alpha_0 \in [0, 1], \quad (6.37a)$$

$$p - p_1 = \sqrt{\rho\rho_1} \sqrt{\frac{p - p_1}{\rho - \rho_1}} (u - u_1), \quad \rho_1 = \rho_1(\alpha_1, p_1), \quad \rho = \rho(\alpha_1, p), \quad \alpha_1 \in [0, 1]. \quad (6.37b)$$

Linearizing  $\sqrt{\frac{p-p_0}{\rho-\rho_0}}$  as  $c_0$ ,  $\sqrt{\frac{p-p_1}{\rho-\rho_1}}$  as  $c_1$ , and taking  $\sqrt{\rho\rho_0}$  and  $\sqrt{\rho\rho_1}$  as  $\rho_0$  and  $\rho_1$ , respectively, more directly gives the familiar linear relations

$$\frac{p - p_0}{u - u_0} = -C_0, \quad C_0 = C_0(\alpha_0, p_0), \quad \alpha_0 \in [0, 1], \quad (6.38a)$$

$$\frac{p - p_1}{u - u_1} = C_1, \quad C_1 = C_1(\alpha_1, p_1), \quad \alpha_1 \in [0, 1]. \quad (6.38b)$$

#### 6.4.4 Intermediate states

For the wave pairs sketched in Figure 6.12, the intermediate states  $(u_{\frac{1}{2}}, p_{\frac{1}{2}})$  lie in or at the boundaries of the shaded quadrilaterals in Figures 6.15a–d. The specific state  $(u_{\frac{1}{2}}, p_{\frac{1}{2}})$  is determined by the specific values  $\alpha_0$  and  $\alpha_1$ . Note that due to the convexity of the Poisson curves, the linearized expressions are more sensitive to cavitation, in case of a double rarefaction, than the nonlinear expressions are. A double rarefaction in pure water is even more sensitive than a double rarefaction in pure air; compare  $(u_{\frac{1}{2}}, p_{\frac{1}{2}})_{\alpha_0=1, \alpha_1=1}$  to  $(u_{\frac{1}{2}}, p_{\frac{1}{2}})_{\alpha_0=0, \alpha_1=0}$  in Figure 6.15a.



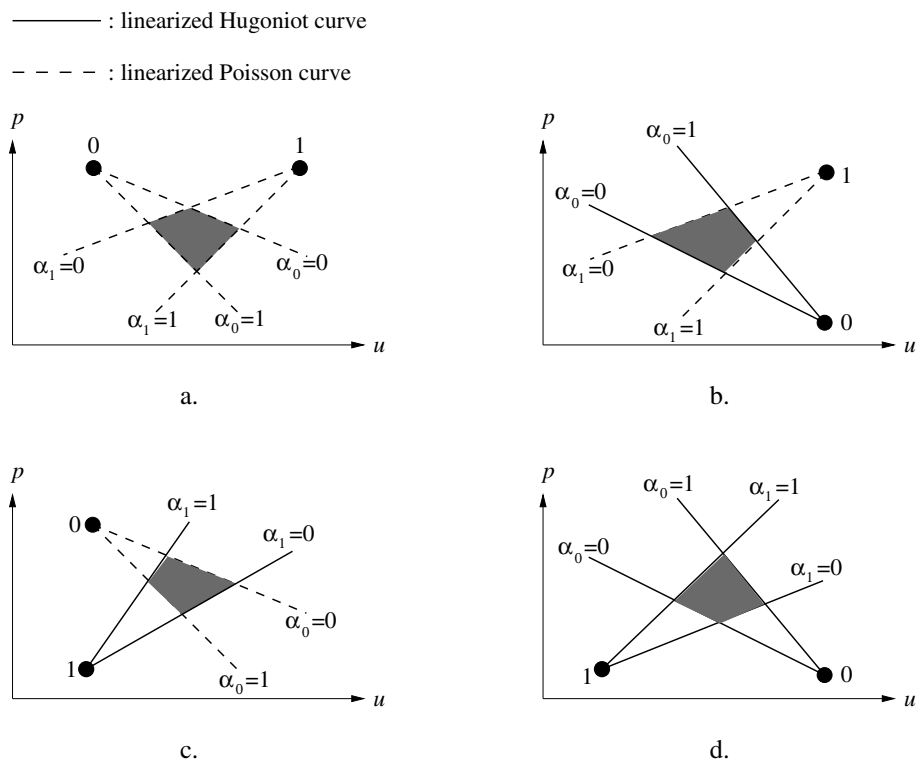


Figure 6.15: Families of linearized Poisson and Hugoniot curves in state space, two-fluid combinations a–d correspond with wave pairs in Figures 6.12a–d.

## 6.5 Conclusions

To accurately compute compressible, immiscible two-fluid flows with very large density differences (water-air flows, e.g.) we have proposed a method that uses a level-set technique to distinguish between the two fluids. The resulting equations have been discretized through a finite-volume method. To compute the fluxes across the finite-volume walls (the level-set flux being one of the flux-vector components), we have proposed a two-fluid, linearized Godunov scheme. The scheme allows a physically correct capturing of the interface across a single cell face, as well as a neat boundary-condition treatment (no sticking of free surfaces to solid walls, e.g.). The scheme combines good physical properties with simplicity and efficiency.

## CHAPTER 7

# Fixes for Solution Errors near Two-Fluid Interfaces

### 7.1 Introduction

A known difficulty of capturing contact discontinuities in a conservative formulation of the two-fluid Euler equations is that zeroth-order solution errors (in literature often referred to as ‘pressure oscillations’) may arise near the contact discontinuity. (For tracking and fitting approaches this problem does not exist.) We will show that the solution error referred to is proportional to the density ratio. For large density jumps across the interface, the errors may even degenerate to instabilities. Fixes for this solution-error problem are available. We refer to the works of Karni [35, 37] and Abgrall [2], their common paper [3], and also to [25, 34, 70]. In most of the available capturing literature though, the ratio of the two densities at the interface is of the order  $1-10^2$ . To our knowledge, only in [25, 70] ratios of approximately  $10^3$ , typical water-air ratios, are considered.

Conservative capturing through the two-fluid, linearized Godunov scheme proposed in the previous chapter also suffers from the problem. In this chapter, we will show this on the basis of a model flow with known exact solution. Flows with a single spatial dimension only are considered. The spatial simplicity allows us to compute fluid-flow problems with known exact solutions.

The contents of this chapter are the following. In Section 7.2, we analyze the solution-error problem near interfaces. Next, in Section 7.3, some approaches to fix the problem are described. Not all of these approaches (some of them already known) appear to work for water-air flow with its large density jump. One fix is proposed which works perfectly, it is a simple variant of the ghost-fluid method [25]. In Section 7.4, numerical results are presented for compressible water-air flows.

## 7.2 Error near interface

### 7.2.1 Fluid-flow equations

The fluid-flow model is identical to the one introduced in Section 6.2. Here we will briefly summarize the equations again. To describe a 1D two-fluid (water-air) flow, consider (for a sufficiently small control volume  $\Omega$ ) the following system of equations:

$$\int_{\Omega} \frac{d}{dt} \begin{pmatrix} \rho \\ \rho u \\ \rho \phi \end{pmatrix} dx + \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u \phi \end{pmatrix}_{\partial\Omega_{right}} - \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u \phi \end{pmatrix}_{\partial\Omega_{left}} = 0, \quad (7.1a)$$

with  $\rho$  the bulk density, which is defined as

$$\rho = \alpha(\phi)\rho_w(p) + (1 - \alpha(\phi))\rho_a(p), \quad (7.1b)$$

where  $\alpha$  is the volume-of-water fraction,  $\phi$  the level-set function, and where  $\rho_w(p)$  and  $\rho_a(p)$  are the equations of state for water and air, respectively. For the latter, we use Tait's:

$$\rho_w(p) = \left( \frac{p + B_w p_{\infty}}{(1 + B_w)p_{\infty}} \right)^{\frac{1}{\gamma_w}} (\rho_w)_{\infty}, \quad \rho_a(p) = \left( \frac{p + B_a p_{\infty}}{(1 + B_a)p_{\infty}} \right)^{\frac{1}{\gamma_a}} (\rho_a)_{\infty}. \quad (7.1c)$$

The level-set function  $\phi$  is initialized as the signed-distance function and is taken positive in water.

### 7.2.2 Analysis of flow model

Consider a 1D tube with unit length,  $x \in [0, 1]$ , inflow at  $x = 0$ , outflow at  $x = 1$  and with as initial solution:

$$u(x, t = 0) = U > 0, \quad (7.2a)$$

$$p(x, t = 0) = P, \quad (7.2b)$$

$$\rho(x, t = 0) = \begin{cases} \rho_w(P), & x \leq (x_{fs})_{t=0}, \\ \rho_a(P), & x \geq (x_{fs})_{t=0}, \end{cases} \quad (7.2c)$$

where  $U$  and  $P$  are constant, and where  $x_{fs}$  is the location of the free surface, the water-air interface. For  $t > 0$ , the corresponding exact Euler-flow solution simply reads  $u(x, t) = U$ ,  $p(x, t) = P$ ,  $\rho(x, t) = \rho_w(P)$  for  $x \leq (x_{fs})_{t=0} + Ut$  and  $\rho(x, t) = \rho_a(P)$  for  $x \geq (x_{fs})_{t=0} + Ut$ , see Figure 7.1. The model flow seems trivial, but it is not. It precisely uncovers the deficiency of capturing methods with regard to material interfaces.

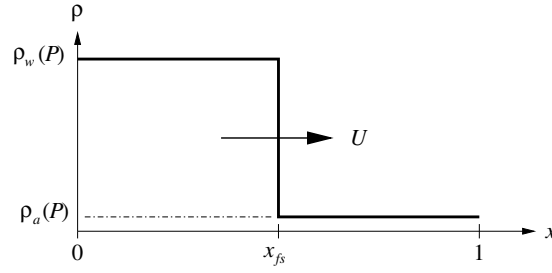


Figure 7.1: Water-air interface running from left to right through a 1D tube, at constant speed  $U$  and pressure  $P$ .

For the problem at hand, for the space discretization of (7.1a) we consider an equidistant finite-volume grid with mesh size  $h$ . For the time integration, just for convenience, we take the forward Euler scheme. The space discretization is taken first-order accurate. Then, denoting the solution in cell  $i$  at the old time level by  $q_i^n$ ,  $q = (\rho, \rho u, \rho \phi)^T$ , we have as equation for the solution  $q_i^{n+1}$  at the new time level:

$$q_i^{n+1} = q_i^n - \frac{\Delta t}{h} (F(q_i^n, q_{i+1}^n) - F(q_{i-1}^n, q_i^n)), \quad (7.3)$$

with  $\Delta t$  the time step, and with  $F$  denoting the two-fluid, linearized Godunov flux (6.22)–(6.23). Considering the situation where  $q_{i-1}^n$ ,  $q_i^n$  and  $q_{i+1}^n$  are according to initial solution (7.2), with  $(x_{fs})^n = x_{i-\frac{1}{2}}$  (see Figure 7.2), with the two-fluid, linearized Godunov scheme we get

$$\rho_i^{n+1} = \sigma \rho_w + (1 - \sigma) \rho_a, \quad (7.4a)$$

$$(\rho u)_i^{n+1} = (\sigma \rho_w + (1 - \sigma) \rho_a) U, \quad (7.4b)$$

$$(\rho \phi)_i^{n+1} = (\sigma \rho_w - (1 - \sigma) \rho_a) \frac{h}{2}, \quad (7.4c)$$

where  $\sigma \equiv \frac{U \Delta t}{h}$  is the CFL number.

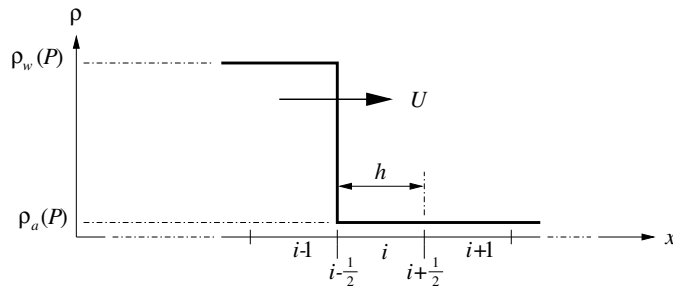


Figure 7.2: Bulk density distribution near cell  $i$  at time level  $n$ , water-air interface at  $x_{i-\frac{1}{2}}$ .

Note that  $\rho_i^{n+1}$  according to (7.4) is exact, as is  $u_i^{n+1} = \frac{(\rho u)_i^{n+1}}{\rho_i^{n+1}} = U$ . However, for  $\phi_i^{n+1}$  it follows from (7.4)

$$\phi_i^{n+1} = \frac{(\rho\phi)_i^{n+1}}{\rho_i^{n+1}} = \frac{\sigma\rho_w - (1-\sigma)\rho_a}{\sigma\rho_w + (1-\sigma)\rho_a} \frac{h}{2}, \quad (7.5)$$

whereas the exact discrete solution reads  $(\phi_i^{n+1})_{exact} = -\frac{h}{2} + \sigma h$ . Hence, for the local discretization error  $\Delta\phi_i^{n+1} = \phi_i^{n+1} - (\phi_i^{n+1})_{exact}$  it holds

$$\Delta\phi_i^{n+1} = \sigma(1-\sigma) \frac{\rho_w - \rho_a}{\sigma\rho_w + (1-\sigma)\rho_a} h. \quad (7.6)$$

So,  $\Delta\phi_i^{n+1} = 0$  only if (trivially)  $\sigma = 0$ , if  $\sigma = 1$ , or if  $\rho_w = \rho_a$ . For  $0 < \sigma < 1$  and  $\rho_w > \rho_a$  it always holds  $\Delta\phi_i^{n+1} > 0$ . The local discretization error (7.6) is  $\mathcal{O}(h)$ , but can *not* be made of higher order by applying a higher-order discretization. This holds for any numerical flux function  $F$ . Higher than first-order accuracy is simply inhibited by the bulk density, which is a smeared out representation of the exact discrete density. Through bulk-density formula

$$\rho(\phi, p) = \alpha(\phi)\rho_w(p) + (1 - \alpha(\phi))\rho_a(p), \quad \alpha(\phi) \in [0, 1], \quad (7.7)$$

with  $\alpha$  denoting the volume-of-fluid (water) fraction, the error (7.6) carries over into a pressure error  $\Delta p_i^{n+1}$ . Given  $\Delta\rho_i^{n+1} = 0$ , from (7.7) it follows after linearization that

$$\begin{aligned} \rho_i^{n+1} &= (\alpha_i^{n+1} + \Delta\alpha_i^{n+1}) \left( \rho_w(P) + \frac{\Delta p_i^{n+1}}{c_w^2(P)} \right) \\ &+ (1 - \alpha_i^{n+1} - \Delta\alpha_i^{n+1}) \left( \rho_a(P) + \frac{\Delta p_i^{n+1}}{c_a^2(P)} \right), \end{aligned} \quad (7.8)$$

hence,

$$\Delta p_i^{n+1} = \frac{-c_w^2 c_a^2}{(\alpha_i^{n+1} + \Delta\alpha_i^{n+1}) c_a^2 + (1 - \alpha_i^{n+1} - \Delta\alpha_i^{n+1}) c_w^2} (\rho_w - \rho_a) \Delta\alpha_i^{n+1}. \quad (7.9)$$

From the formulas for the volume-of-fluid fraction  $\alpha_i$  as given in (6.12)–(6.13), it follows that, besides on  $\Delta\phi_i^{n+1}$ ,  $\Delta\alpha_i^{n+1}$  also depends on  $\Delta\phi_{i-1}^{n+1}$  and  $\Delta\phi_{i+1}^{n+1}$ . For the model flow considered, it follows with the two-fluid, linearized Godunov scheme:  $\phi_{i-1}^{n+1} = \frac{(\rho\phi)_{i-1}^{n+1}}{\rho_{i-1}^{n+1}} = \frac{h}{2} + \sigma h$  and  $\phi_{i+1}^{n+1} = \frac{(\rho\phi)_{i+1}^{n+1}}{\rho_{i+1}^{n+1}} = -\frac{3h}{2} + \sigma h$ , which are both the exact results. Resuming, we have  $\phi_i^{n+1} = -\frac{h}{2} + \sigma h + \Delta\phi_i^{n+1}$ ,  $\phi_{i-\frac{1}{2}}^{n+1} = \sigma h + \frac{1}{2}\Delta\phi_i^{n+1}$  and  $\phi_{i+\frac{1}{2}}^{n+1} = -h + \sigma h + \frac{1}{2}\Delta\phi_i^{n+1}$ , with  $\Delta\phi_i^{n+1}$  according to (7.6). With the procedure for computing the volume-of-fluid fraction  $\alpha$  as described in Section 6.3.1, the following expressions can then be derived for the error  $\Delta\alpha_i^{n+1} = \alpha_i^{n+1} - (\alpha_i^{n+1})_{exact}$ :

$$\Delta\alpha_i^{n+1} = \left( \frac{1}{2} + \sigma \right) \frac{\Delta\phi_i^{n+1}}{h - \Delta\phi_i^{n+1}}, \quad \phi_i^{n+1} \leq 0, \quad (7.10a)$$

$$\Delta\alpha_i^{n+1} = \left(\frac{3}{2} - \sigma\right) \frac{\Delta\phi_i^{n+1}}{h + \Delta\phi_i^{n+1}}, \quad \phi_i^{n+1} \geq 0. \quad (7.10b)$$

Note that  $\Delta\alpha_i^{n+1} = \mathcal{O}(1)$ , i.e., mesh-size-independent and – hence – with (7.9) also the pressure error  $\Delta p_i^{n+1}$  is! Also note that the pressure error (7.9) is proportional to the density ratio  $\frac{\rho_w}{\rho_a}$ . This illustrates the poor, density-ratio-dependent solution-error behavior already mentioned in the beginning of this chapter. Note that the error behavior is not so poor that the denominator  $h - \Delta\phi_i^{n+1}$  in (7.10a) may become zero for  $0 < \sigma < 1$  and  $\rho_w > \rho_a$ .

### 7.2.3 Guidelines for error improvements

Before proposing improvements of the poor local error behavior near the interface, it is useful to make an error analysis of bulk-density relation (7.7) and – also – to consider the conservative equations near the interface.

#### Error analysis of bulk-density relation

Errors in the pressure and volume-of-fluid fraction ( $\Delta p$  and  $\Delta\alpha$ ) induce an error in the bulk density ( $\Delta\rho$ ), which, given (7.7), satisfies the equation

$$\rho + \Delta\rho = (\alpha + \Delta\alpha)\rho_w(p + \Delta p) + (1 - \alpha - \Delta\alpha)\rho_a(p + \Delta p). \quad (7.11)$$

For the model flow and discretization method considered in Section 7.2.2, we found  $\Delta\rho = 0$  and  $\Delta\alpha = \mathcal{O}(1)$ . Then, according to (7.11),  $\Delta p = \mathcal{O}(1)$  as well, which is in agreement with what we derived in Section 7.2.2. Near the interface, instead of the zeroth-order pressure error  $\Delta p$  described by (7.9)–(7.10), we ideally prefer  $\Delta p = 0$ , which implies according to (7.11)

$$\Delta\rho = \Delta\alpha(\rho_w(p) - \rho_a(p)). \quad (7.12)$$

One of the fixes to be considered in the following is to make the numerical method so that  $\Delta\rho$  and  $\Delta\alpha$  exactly satisfy (7.12).

#### Reconsideration of fluid-flow equations near interface

Consider the situation in which the interface is in cell  $\Omega = \Omega_i$  (*only* the interface, so no shock or rarefaction). Since velocity and pressure are continuous across the interface, for sufficiently small  $\Omega_i$ , we may then write by good approximation:  $u_{i-\frac{1}{2}} = u_{i+\frac{1}{2}} = u_i$  and  $p_{i-\frac{1}{2}} = p_{i+\frac{1}{2}}$ . With this, (7.1a) can be rewritten as

$$\int_{\Omega_i} \frac{d}{dt} \begin{pmatrix} \rho \\ \rho u \\ \rho\phi \end{pmatrix} dx + u_i \left( \begin{pmatrix} \rho \\ \rho u \\ \rho\phi \end{pmatrix}_{\partial\Omega_{i+\frac{1}{2}}} - \begin{pmatrix} \rho \\ \rho u \\ \rho\phi \end{pmatrix}_{\partial\Omega_{i-\frac{1}{2}}} \right) = 0, \quad (7.13)$$

i.e., as

$$\int_{\Omega_i} \frac{dq}{dt} dx + u_i (q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}}) = 0, \quad (7.14)$$

which is a system of advection equations for the entire solution vector  $q_i$ . (Contact discontinuities are linear phenomena.) If all conservative solution components are advected –  $\rho_i$ ,  $(\rho u)_i$  and  $(\rho\phi)_i$  – then any solution component in  $\Omega_i$ , either conservative or non-conservative, is. I.e., in (7.14), in a cell with (only) a contact discontinuity, instead of the fully conservative solution representation  $q_i = (\rho_i, (\rho u)_i, (\rho\phi)_i)$  we may equally well consider, e.g., the partially conservative representation  $q_i = (\rho_i, (\rho u)_i, \phi_i)$ , the fully non-conservative representation  $q_i = (u_i, p_i, \phi_i)$ , or whatever. This knowledge is important. In combination with error equation (7.11), it allows us to derive a fix for the zeroth-order error observed in Section 7.2.2.

### 7.3 Fixes for error in cell with interface

#### 7.3.1 Advection of level-set function

This approach is based on the observation made in Section 7.2.2 that the update of  $\phi_i^n$  through division of  $(\rho\phi)_i^{n+1}$  by  $\rho_i^{n+1}$  leads to a first-order accuracy barrier in  $\phi_i^{n+1}$  because of the intrinsic smearing in the bulk-density representation itself. For the update of the real physical quantities  $\rho_i$  and  $(\rho u)_i$  we may stick to the conservative formulation and, hence, to the two-fluid, linearized Godunov scheme. Doing so, with the forward Euler, first-order upwind discretization of the single advection equation

$$\int_{\Omega_i} \frac{d\phi}{dt} dx + u_i \left( \phi_{i+\frac{1}{2}} - \phi_{i-\frac{1}{2}} \right) = 0, \quad (7.15)$$

for the model flow considered – in addition to (7.4a) and (7.4b) for  $\rho_i^{n+1}$  and  $(\rho u)_i^{n+1}$  – we get

$$\phi_i^{n+1} = -\frac{h}{2} + \sigma h, \quad (7.16)$$

which is exact. Note that  $\phi_i^{n+1}$  is exact because  $\phi$  has been defined as the signed-distance function. A nonlinear spatial distribution of  $\phi$  would have yielded an error  $\Delta\phi_i^{n+1}$ . Because  $\Delta\phi_i^{n+1} = 0$ , it also holds  $\Delta\alpha_i^{n+1} = 0$ . Since  $\Delta\rho_i^{n+1} = 0$  as well (Section 7.2.2), from (7.11) it then follows  $\Delta p_i^{n+1} = 0$ . In all other cells, the fully conservative scheme (7.3) is applied, yielding there the exact discrete solution. However, at time level  $n+2$  the numerical solution is no longer exact. According to the linearized Godunov scheme it holds for first-order state interpolation to the cell faces:  $\rho_{i-\frac{1}{2}}^{n+1} = \rho_{i-1}^{n+1} = \rho_w$ ,  $\rho_{i+\frac{1}{2}}^{n+1} = \rho_i^{n+1} = \sigma\rho_w + (1-\sigma)\rho_a$  and  $u_{i-\frac{1}{2}}^{n+1} = u_{i+\frac{1}{2}}^{n+1} = U$ . Then, from (7.3) it follows  $\rho_i^{n+2} = 2\sigma\rho_w + (1-2\sigma)\rho_a - \sigma^2(\rho_w - \rho_a)$ , whereas  $(\rho_i^{n+2})_{exact} = 2\sigma\rho_w + (1-2\sigma)\rho_a$ . Hence,  $\Delta\rho_i^{n+2} = -\sigma^2(\rho_w - \rho_a)$ . With the forward-Euler, first-order upwind discretization of (7.15), it follows  $\Delta\phi_i^{n+2} = 0$  and, as a consequence,  $\Delta\alpha_i^{n+2} = 0$ . With (7.11), it then follows, since  $\Delta\rho_i^{n+2} = \mathcal{O}(1)$ , that  $\Delta p_i^{n+2} = \mathcal{O}(1)$ . So, this partially conservative approach is not a fix. With some tricks one can make the method work. Taking for the left and right cell-face densities to be substituted into the linearized Godunov scheme, instead of the *bulk* densities,



the *local (non-bulk)* densities (pure water or pure air) at  $t = t^n$ , the method works as long as the interface does not cross a cell face during a time step. I.e., the method works for  $\sigma = \frac{1}{m}$ , with  $m$  integer. For the problem at hand, verify that, instead of  $\rho_{i+\frac{1}{2}}^{n+1} = \sigma\rho_w + (1-\sigma)\rho_a$ , this would have yielded  $\rho_{i+\frac{1}{2}}^{n+1} = \rho_a$  (pure air), and so (with  $\rho_{i-\frac{1}{2}}^{n+1} = \rho_w$  and  $u_{i\pm\frac{1}{2}}^{n+1} = U$ ):  $\rho_i^{n+2} = 2\sigma\rho_w + (1-2\sigma)\rho_a$ , thus  $\Delta\rho_i^{n+2} = 0$ , and – hence – with  $\Delta\alpha_i^{n+2} = 0$ :  $\Delta p_i^{n+2} = 0$ , instead of  $\Delta p_i^{n+2} = \mathcal{O}(1)$ . However, aforementioned requirement on  $\sigma$  is too restrictive to let the method be of much practical use.

### 7.3.2 Advection of velocity, pressure and level-set function

Taking in (7.14)  $q_i = (u_i, p_i, \phi_i)$ , with  $(u_{i-\frac{1}{2}}, p_{i-\frac{1}{2}}) = (u_{i+\frac{1}{2}}, p_{i+\frac{1}{2}})$ , it follows the exact result

$$\begin{pmatrix} u \\ p \\ \phi \end{pmatrix}_i^{n+1} = \begin{pmatrix} u \\ p \\ \phi \end{pmatrix}_i^n + \begin{pmatrix} 0 \\ 0 \\ \sigma h \end{pmatrix} = \begin{pmatrix} U \\ P \\ -\frac{h}{2} + \sigma h \end{pmatrix}. \quad (7.17)$$

However, in cell  $\Omega_{i+1}$  an error arises. Verify that  $q_{i+1}^{n+1}$  is still exact,  $q_{i+1}^{n+1} = (U, P, -\frac{3}{2}h + \sigma h)$ , as is  $q_{i+2}^{n+1} = (U, P, -\frac{5}{2}h + \sigma h)$ , but for  $t = t^{n+2}$  we find with the linearized Godunov scheme:  $\rho_{i+1}^{n+2} = \rho_a + \sigma^2(\rho_w - \rho_a)$ . For  $\sigma < \frac{1}{2}$ , this is wrong; water is erroneously transported from cell  $i$  into cell  $i+1$ . The corresponding error reads:  $\Delta\rho_i^{n+2} = \sigma^2(\rho_w - \rho_a) = \mathcal{O}(1)$ . Meanwhile, for  $\sigma < \frac{1}{2}$  such that  $\phi_{i+\frac{1}{2}}^{n+2} = \phi_{i+\frac{3}{2}}^{n+2}$  are both still negative, we correctly find  $\alpha_{i+1}^{n+2} = 0$ . So, with (7.11) it then follows  $\Delta p_{i+1}^{n+2} = \mathcal{O}(1)$  and therefore this approach – although (trivially) fixing the pressure-error problem in the cell with interface – is not (yet) good either, because it yields an error in a neighbouring cell.

### 7.3.3 Advection of density and volume-of-fluid fraction

In Section 7.2.3 we have seen that if  $\Delta\rho$  and  $\Delta\alpha$  are such that (7.12) is satisfied, then  $\Delta p = 0$ . We derive a possible fix which is based on (7.12). On the basis of general advection equation (7.14), we can directly write the advection equations

$$\int_{\Omega_i} \frac{d\rho}{dt} dx + u_i \left( \rho_{i+\frac{1}{2}} - \rho_{i-\frac{1}{2}} \right) = 0, \quad (7.18a)$$

$$\int_{\Omega_i} \frac{d\alpha}{dt} dx + u_i \left( \alpha_{i+\frac{1}{2}} - \alpha_{i-\frac{1}{2}} \right) = 0. \quad (7.18b)$$

The peculiar cell-face based volume-of-fluid fractions  $\alpha_{i-\frac{1}{2}}$  and  $\alpha_{i+\frac{1}{2}}$  in (7.18b) may become functions of real volume-of-fluid fractions upon further discretization. Yet, further discretization is not necessary since (7.18a) and (7.18b) are identical. ‘Extension’ of the relation  $p_{i-\frac{1}{2}} = p_{i+\frac{1}{2}}$ , which underlies (7.14), to  $p_{i-\frac{1}{2}} = p_{i+\frac{1}{2}} = p_i$  implies that with  $\rho = \alpha\rho_w(p) + (1-\alpha)\rho_a(p)$ , (7.18a) can be rewritten as (7.18b).

So, when we maintain (7.18a), the updates  $(\rho_i^{n+1} - \rho_i^n)$ ,  $(\rho_i^{n+2} - \rho_i^{n+1})$ , etc. which it renders, may be directly translated (through (7.12)) into updates  $(\alpha_i^{n+1} - \alpha_i^n)$ ,  $(\alpha_i^{n+2} - \alpha_i^{n+1})$ , etc. Whereas in the previous fix the pressure error was explicitly set to zero, here it is implicitly done so. However, as with the fully non-conservative approach from Section 7.3.2, in the second time step an  $\mathcal{O}(1)$  pressure error arises in neighbouring cell  $\Omega_{i+1}$ , when the fully conservative approach is still applied there. Moreover, even in a better case, this fix will yield an exact pressure solution at the expense of a diffused density profile.

A perfect fix is a variant of the so-called ghost-fluid method [25]. This variant is described in the next section.

### 7.3.4 Ghost-fluid method

In [25], the ghost-fluid method is introduced for the non-homentropic Euler equations of gas dynamics. For our more compact system of fluid-flow equations, we propose a simple variant of the ghost-fluid method. As the so-called ghost cells we define those cells in which there is an interface, i.e., a zero of the level-set function. These cells are considered in an ambiguous manner: as fully filled with water (ghost water) and as fully filled with air (ghost air). Then, still considering the 1D situation for convenience, across the two walls of the ghost cell, both water and air fluxes of mass and momentum are computed (ghost fluxes). On the basis of the difference between the two ghost-water fluxes, the ghost-water solution is updated, i.e., its mass and momentum. Likewise, the ghost-air solution is. Expressed in  $(u, p)$ -variables, these two new ghost solutions (for water and air) will be exactly the same in the absence of pressure waves, which is the case for the 1D problem introduced in Section 7.2.2. Throughout the entire computational domain the level-set function is simply advected with (7.15). In case the updated ghost-water and ghost-air solutions do differ, we propose the following. From the (updated) level-set solution, the volume-of-fluid fraction in the ghost cell can be computed. Then, the solution in the ghost cell is made unique with

$$\begin{pmatrix} u \\ p \end{pmatrix} = \alpha \begin{pmatrix} u_w \\ p_w \end{pmatrix} + (1 - \alpha) \begin{pmatrix} u_a \\ p_a \end{pmatrix}. \quad (7.19)$$

There are no physical or mathematical arguments for applying this weighting, other choices are possible. The fluxes (real and ghost) are computed with the single-fluid version of the two-fluid, linearized Godunov scheme. To compute a water flux (either real or ghost), in the expressions for  $u_{\frac{1}{2}}$  and  $p_{\frac{1}{2}}$  given in Section 6.3.3, for  $\rho_0, \rho_1, c_0$  and  $c_1$  the water values are taken. The computations of the air fluxes is done in a similar fashion. Note that in the computation of all types of fluxes (real or ghost, water or air), use is made of the same, unique values of  $u$  and  $p$  in each cell. This uniqueness ensures that the free-surface conditions are satisfied implicitly. In [25], for the non-homentropic Euler equations of gas dynamics, entropy is extrapolated across the interface. The present homentropic equations do not require any solution-component extrapolation. Tangential velocity components do not yet apply here. In multi-D, in each cell, besides the physical

normal velocity component and pressure, we would also use the physical tangential velocity components available there. With Navier-Stokes as the ultimate flow model, like the normal velocity component and pressure, the tangential velocity components will also be continuous across the interface. Considering the 1D situation with, at time  $t = t^n$ , the interface somewhere in cell  $i$ , with – say – water at the left, in case of the first-order accurate space discretization, the fluxes to be computed are those depicted in Figure 7.3. So, only across the cell faces  $i - \frac{1}{2}$  and  $i + \frac{1}{2}$  ghost-water and ghost-air fluxes are computed, across all cell faces left of  $i - \frac{1}{2}$ : real water fluxes and across all cell faces right of  $i + \frac{1}{2}$ : real air fluxes.

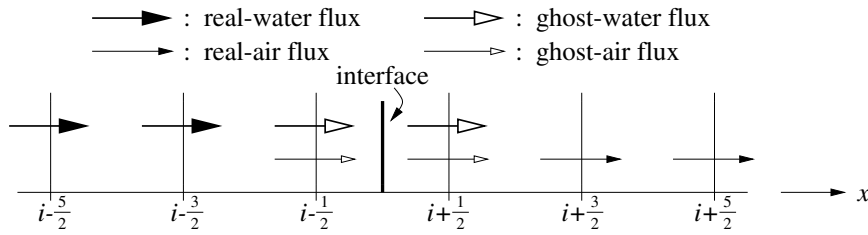


Figure 7.3: Types of fluxes computed in present ghost-fluid method.

Note that in the ghost-fluid method the interface is no longer captured at the smallest discrete level (that of a cell face), but at the next larger: a cell. Because in the ghost-fluid method fluxes are always of the single-fluid type, explicit calculation of  $u_{\frac{1}{2}}$  and  $p_{\frac{1}{2}}$  can be done by using, e.g., Osher's scheme, instead of the two-fluid, linearized Godunov scheme.

## 7.4 Numerical results

### 7.4.1 Water front at constant speed and pressure

#### Numerical choices

The first test case to be considered is the 1D tube flow already introduced in Section 7.2.2. Numerical values to be considered are:  $(x_{fs})_{t=0} = 0.5$  (initial interface halfway tube) or  $(x_{fs})_{t=0} = 0$  (initial interface at inlet boundary),  $U = 1$ ,  $P = 1$ ,  $\rho_w(P) = 1$ ,  $\rho_a(P) = 0.001$  (in the ideal case),  $\gamma_w = 7$ ,  $\gamma_a = \frac{7}{5}$ ,  $B_w = 3000$  and  $B_a = 0$ . According to the speed-of-sound relations

$$c_w^2 = \gamma_w \frac{(1 + B_w)p}{\rho_w}, \quad (7.20a)$$

$$c_a^2 = \gamma_a \frac{p}{\rho_a}, \quad (7.20b)$$

these values imply  $c_w(P) \approx \sqrt{15} c_a(P)$ , which agrees fairly well with common sea-level conditions. As in Section 7.2, the grids to be used are equidistant. The boundary conditions to be imposed are  $u(x = 0, t) = U > 0$ ,  $\phi(x = 0, t) = Ut$  and  $p(x = 1, t) = P$ . The space discretization is taken first-order accurate, like in Section 7.2.2. Time integration is done with the forward Euler scheme, with the time step constant and sufficiently small to guarantee stability:

$$\Delta t = \sigma \frac{h}{U + c_w(P)}, \quad \sigma < 1. \quad (7.21)$$

#### Results fully conservative approach

This is the approach without any fixes for solution errors near the interface. For the numerical values just mentioned, the computation breaks down. Stumbling block is the large density ratio. In Figure 7.4, pressure errors are depicted for computations with successively the following three still rather small density ratios:  $\frac{\rho_w}{\rho_a} = 2, 4$  and  $8$ , and after the following three numbers of time steps: 10 (left column of graphs), 20 (middle column of graphs) and 40 (right column). The time step on the coarsest grid is twice as large as that on the middle grid and four times larger than that on the finest grid. So, note that in each of the eight graphs (no results were obtained for  $\frac{\rho_w}{\rho_a} = 8$  and 40 time steps), the three pressure-error distributions correspond with the same number of time steps (10, 20 or 40), *not* with the same time. The pressure error appears to be about linearly proportional to the number of time steps taken. In agreement with the theoretical findings, it also increases with the density ratio  $\frac{\rho_w}{\rho_a}$ . The latter increase is clearly nonlinear. With the conservative approach, results for  $\frac{\rho_w}{\rho_a} = 1000$  are still far out of reach. The deceptive performance of the conservative approach was expected given the analytical results of Section 7.2.2.

#### Results advection of level-set function

Here the fix proposed in Section 7.3.1 is numerically investigated. The fix is applied not only in the cell in which the interface actually is, but also in its left and right

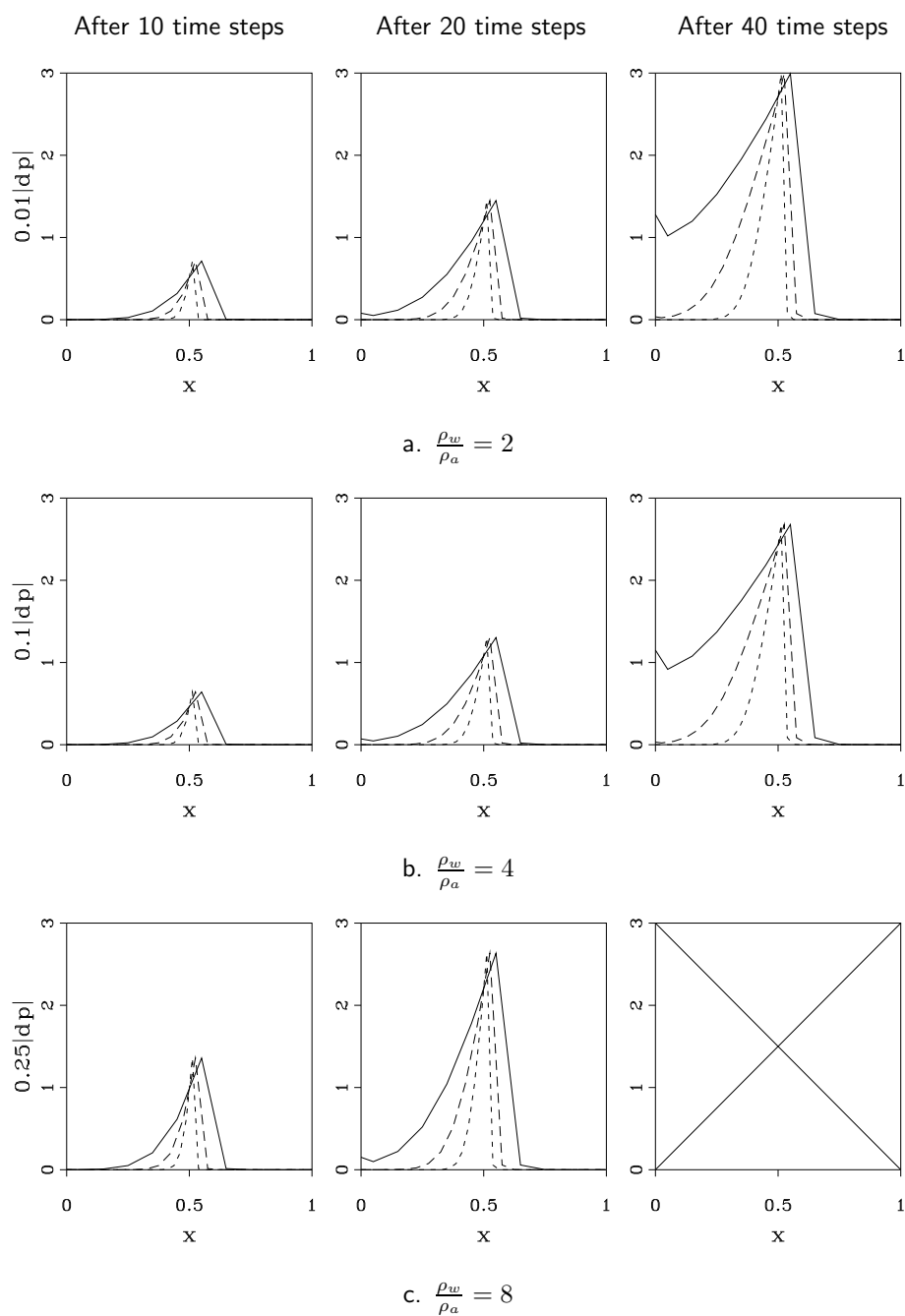


Figure 7.4: Pressure-error distributions fully conservative approach (solid lines:  $h = \frac{1}{10}$ , coarsely dashed lines:  $h = \frac{1}{20}$ , finely dashed lines:  $h = \frac{1}{40}$ ).

neighbouring cell. The fix clearly gives an improvement as compared to the fully conservative approach, but the fix is not adequate. For  $(x_{fs})_{t=0} = 0$  and  $\frac{\rho_w}{\rho_a} = 10$ , in each of the three graphs in Figure 7.5 we present the computed bulk-density profiles at  $t = 0.0, 0.1, 0.2, \dots, 1.0$ . The results look perfect, but they are not. They are cursed with a pressure error, which for  $\frac{\rho_w}{\rho_a} = 10$  is still negligibly small. But, as in the previous section, the error grows rapidly with increasing density ratio  $\frac{\rho_w}{\rho_a}$ . Results similar to those in Figure 7.5 cannot be obtained for  $\frac{\rho_w}{\rho_a} = 1000$ , not even for  $\frac{\rho_w}{\rho_a} = 100$ . From Figure 7.6 it appears that the pressure error grows exponentially with  $\frac{\rho_w}{\rho_a}$ .

### Results advection of velocity, pressure and level-set function

Here, the fix proposed in Section 7.3.2 is tested. The advection of  $u, p$  and  $\phi$  is applied in the cell with interface as well as in its left and right neighbouring cell. The fix is an improvement compared to that with advection of  $\phi$  only, but it does not work satisfactorily either. It also breaks down for increasing density ratio  $\frac{\rho_w}{\rho_a}$ ; for  $\frac{\rho_w}{\rho_a} = 100$  after  $t = 0.7$ , and for  $\frac{\rho_w}{\rho_a} = 1000$  after  $t = 0.4$ , see Figure 7.7.

### Results ghost-fluid method

The fix proposed in Section 7.3.3 is skipped, its expected smearing of the density excludes it as an interesting option here. The ghost-fluid method described in Section 7.3.4 is interesting; it works, see Figure 7.8. For the problem at hand, it even works for arbitrarily large density ratios, as can be seen in Figure 7.9.

## 7.4.2 Oscillating water column

### Analysis

Although the previous constant-speed-and-pressure test case is not trivial from a numerical point of view, from a physical perspective it is. As a second test case we propose the following, physically more interesting test case. Consider a 1D tubular circuit with a valve in it which for  $t < 0$  is open. The tube contains a water and air column (Figure 7.10), flowing (for  $t < 0$ ) at constant speed  $U$  and pressure  $P$ .

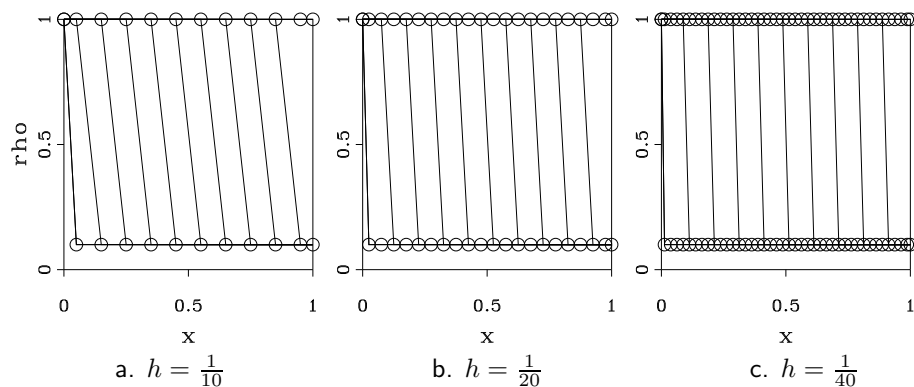


Figure 7.5: Bulk-density profiles at  $t = 0.0, 0.1, 0.2, \dots, 1.0$ , fix with advection of level-set function,  $\frac{\rho_w}{\rho_a} = 10$ .

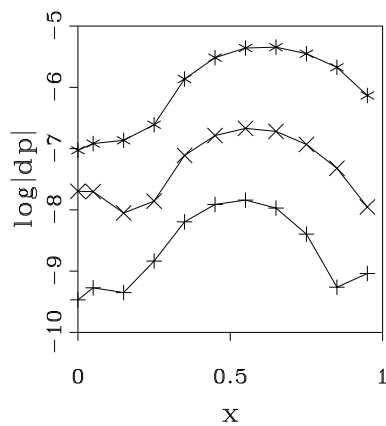


Figure 7.6: Pressure-error distributions at  $t = 0.1$  for  $\frac{\rho_w}{\rho_a} = 70$  (+),  $75$  (x) and  $80$  (\*), fix with advection of level-set function,  $h = \frac{1}{40}$ .

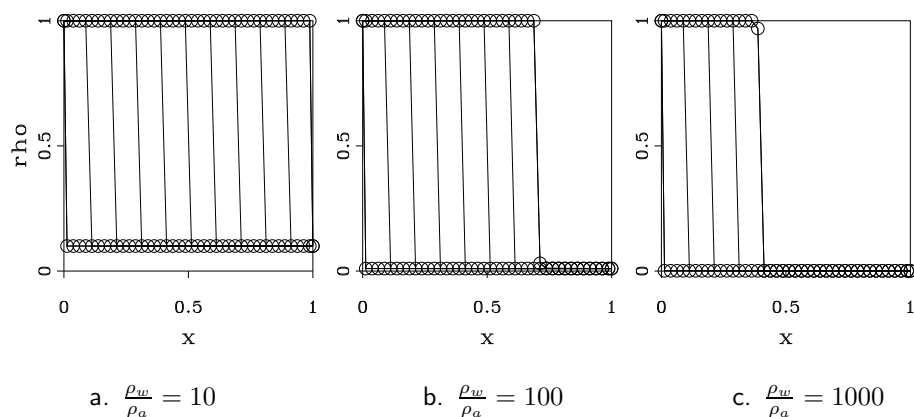


Figure 7.7: Bulk-density profiles at  $t = 0.0, 0.1, 0.2, \dots, 1.0$ , fix with advection of velocity, pressure and level-set function,  $h = \frac{1}{40}$ .

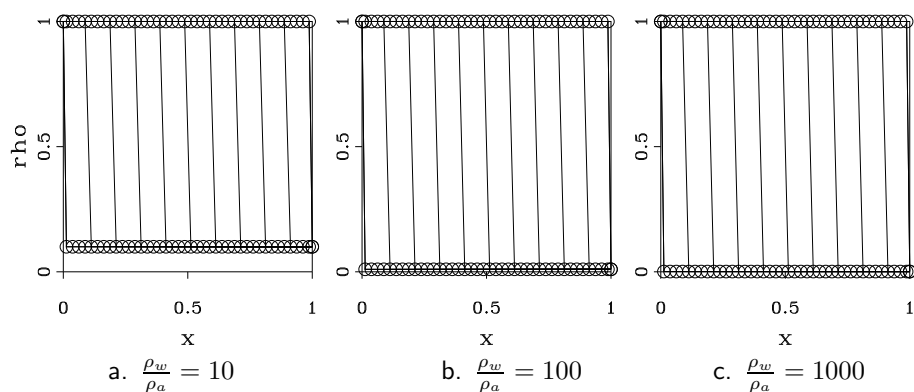


Figure 7.8: Bulk-density profiles at  $t = 0.0, 0.1, 0.2, \dots, 1.0$ , ghost-fluid method,  $h = \frac{1}{40}$ .

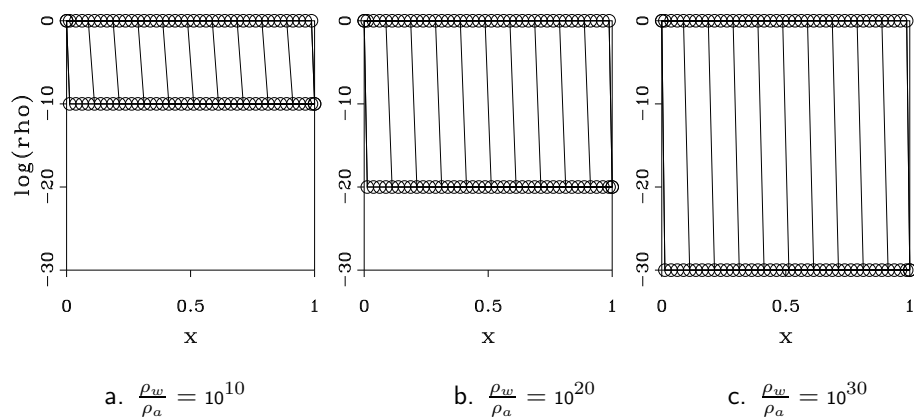


Figure 7.9: Bulk-density profiles at  $t = 0.0, 0.1, 0.2, \dots, 1.0$ , ghost-fluid method, extremely high density ratios,  $h = \frac{1}{40}$ .



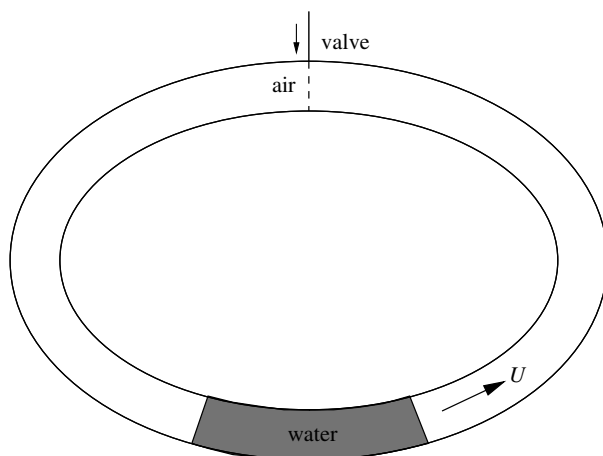


Figure 7.10: Tubular circuit with columns of water and air flowing at constant speed  $U$  and pressure  $P$ .

At  $t = 0$ , when the water column's center is at the maximum distance from the valve, the latter is instantaneously closed. Ignoring curvilinearity, the initial situation is as sketched in Figure 7.11.

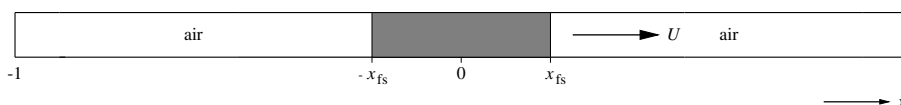


Figure 7.11: Initial condition: shut off tube with column of water in between two columns of air, all three columns flowing to the right at constant speed  $U$  and pressure  $P$ .

Then, starting from  $t = 0$ , the air at the right will be compressed by the water and the air at the left will expand. Hence, a pressure difference is built up across the column of water, with as a consequence a deceleration of the latter's flow to the right, followed by a stagnation, and next an acceleration and flow to the left. The latter leads to a reverse pressure gradient across the water, which will redirect the flow from left to right again, and so on. Hence an oscillatory flow will emerge. A subtle difficulty of this flow problem as compared to the foregoing translating water-air interface is that here pressure waves interact with the water-air interfaces, whereas in the foregoing translating water-air interface case, there are no pressure waves at all.

An elementary analytical flow solution can be derived by making three simplifying assumptions. The first is that the density in the two air columns depends on time  $t$  only (not on the spatial coordinate  $x$ ). Then, denoting the displacement of the water column in positive  $x$ -direction by  $s(t)$ , the density in the left and right air

columns can be written as

$$\rho_a^l(t) = \frac{1 - x_{fs}}{1 - x_{fs} + s(t)} \rho_a(0) \quad \text{and} \quad (7.22a)$$

$$\rho_a^r(t) = \frac{1 - x_{fs}}{1 - x_{fs} - s(t)} \rho_a(0), \quad (7.22b)$$

respectively. With the equation of state, for the pressure in the two air columns it then follows:

$$p^l(t) = \left( \frac{1 - x_{fs}}{1 - x_{fs} + s(t)} \right)^{\gamma_a} P, \quad (7.23a)$$

$$p^r(t) = \left( \frac{1 - x_{fs}}{1 - x_{fs} - s(t)} \right)^{\gamma_a} P. \quad (7.23b)$$

The second assumption is that the water column behaves as a rigid body. With this, Newton's second law of motion applied to the water column's motion reads

$$\rho_w 2x_{fs} \frac{d^2 s}{dt^2} = p^l(t) - p^r(t), \quad (7.24)$$

which, with (7.23a) and (7.23b), yields as differential equation for  $s(t)$ :

$$\rho_w 2x_{fs} \frac{d^2 s}{dt^2} = \left( \left( \frac{1 - x_{fs}}{1 - x_{fs} + s(t)} \right)^{\gamma_a} - \left( \frac{1 - x_{fs}}{1 - x_{fs} - s(t)} \right)^{\gamma_a} \right) P. \quad (7.25)$$

The third assumption made is that the water column's displacements are small with respect to the length of the initial air columns:  $|s(t)| \ll 1 - x_{fs}$ . With this, the pressure expressions (7.23a) and (7.23b), and hence the nonlinear differential equation (7.25), can be linearized by good approximation:

$$\frac{d^2 s}{dt^2} + \omega^2 s = 0, \quad \omega^2 = \frac{\gamma_a P}{\rho_w x_{fs} (1 - x_{fs})}. \quad (7.26)$$

With the initial conditions  $s(0) = 0$  and  $\frac{ds(0)}{dt} = U$ , the solution of (7.26) is

$$s(t) = \frac{U}{\omega} \sin \omega t; \quad (7.27)$$

the water column makes a harmonic oscillation with amplitude  $\frac{U}{\omega}$  and period  $\tau = \frac{2\pi}{\omega}$ . The assumption that the pressure in the air columns is space-independent is satisfied by good approximation if  $c_a \tau \gg 1 - x_{fs}$ , i.e., if

$$2\pi \sqrt{\frac{x_{fs}}{1 - x_{fs}} \frac{\rho_w}{\rho_a(P)}} \gg 1. \quad (7.28a)$$

The small-displacement assumption  $|s(t)| \ll 1 - x_{fs}$  implies

$$\frac{U}{c_a(P)} \sqrt{\frac{x_{fs}}{1 - x_{fs}} \frac{\rho_w}{\rho_a(P)}} \ll 1. \quad (7.28b)$$

Requirements (7.28a) and (7.28b) conflict easily. Assuming that (7.28a) is satisfied, (7.28b) can only be satisfied if  $\frac{U}{c_a(P)} \ll 1$ , i.e., low Mach-number air flow. With the linearized versions of pressure relations (7.23a) and (7.23b), for the pressure coefficients  $\mathcal{P}^l \equiv \frac{p^l(t)-P}{P}$  and  $\mathcal{P}^r \equiv \frac{p^r(t)-P}{P}$ , it holds

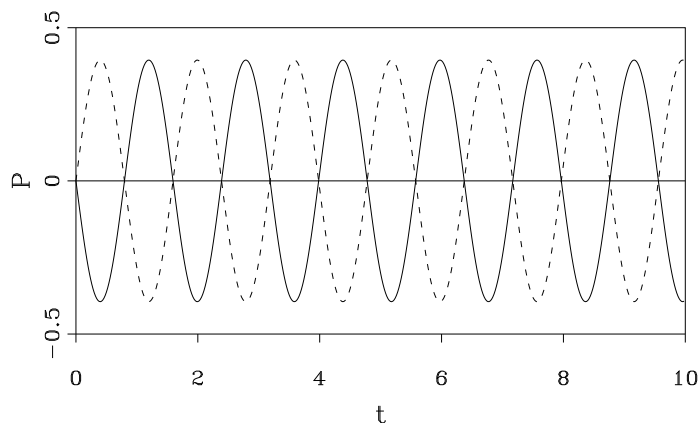
$$\mathcal{P}^l(t) = -\gamma_a \frac{s(t)}{1 - x_{fs}}, \quad (7.29a)$$

$$\mathcal{P}^r(t) = \gamma_a \frac{s(t)}{1 - x_{fs}}. \quad (7.29b)$$

The two coefficients are exactly opposite in phase.

### Numerics

We proceed by presenting numerical results obtained through the ghost-fluid method. In the numerical computations, both water and air are taken as compressible. As for the previous test case, we take  $\gamma_w = 7$ ,  $\gamma_a = \frac{7}{5}$ ,  $B_w = 3000$ ,  $B_a = 0$ ,  $\rho_w(P) = 1$  and  $\rho_a(P) = 0.001$ . Furthermore, we take  $U = 1$ ,  $P = 1$  and  $x_{fs} = 0.1$ . For these numerical choices, in Figure 7.12a we first give the time evolution of  $\mathcal{P}^l$  and  $\mathcal{P}^r$  according to the analytical estimates (7.29a) and (7.29b). For the numerical computations, an equidistant grid with  $h = \frac{1}{40}$  is applied. The space discretization is again first-order accurate and time integration is once more done with the forward Euler scheme. The level-set function is taken as the signed-distance function. For this test case, as opposed to the foregoing, the level-set function is reinitialized. The reinitialization is done after each time step and is done as a signed-distance function. In Figure 7.12b the time evolution of the pressure coefficients  $\mathcal{P}(x = -1, t) = \frac{p(x=-1, t)-P}{P}$  and  $\mathcal{P}(x = 1, t) = \frac{p(x=1, t)-P}{P}$  is given. Note that although requirement (7.28b) is not satisfied very well for the aforementioned numerical values, the analytical results depicted in Figure 7.12a still agree fairly well with the nonlinear numerical results. Also for this test case, the ghost-fluid method appears to work fine.



a. According to analysis

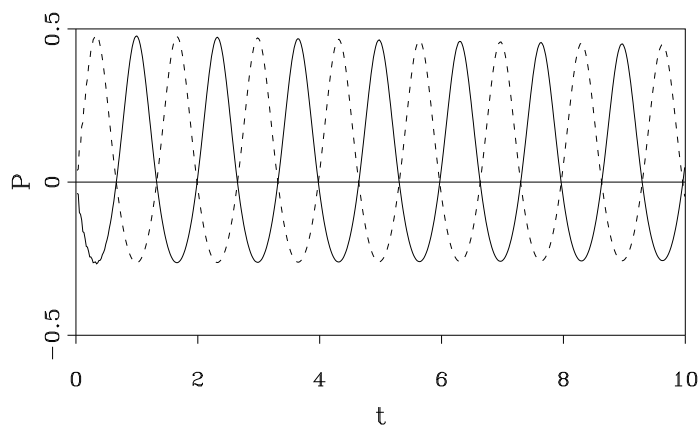
b. According to numerical computations,  $h = \frac{1}{40}$ .

Figure 7.12: Time evolution of pressure coefficients at left and right boundary (solid lines: left boundary, dashed lines: right boundary).

In the ghost cells, the conservation laws are applied to virtual (ghost) amounts of water and air, not to the real physical amounts. So, conservation of the real amounts of mass and momentum is not guaranteed automatically. In case of accurate resolution of the level-set function and the pressure, in and near the ghost cell(s), mass and momentum are expected to be accurately conserved though (*not* exactly). In Figure 7.13a we give the time evolution of the relative mass error  $\mathcal{M}(t) \equiv \frac{m_a(t) - m_a(0)}{m_a(0)}$ , where  $m_a(t)$  is the total mass of air in the closed tube at time  $t$ . The mass error appears to be composed of two components: one oscillating and the other behaving linearly in time. Both obey the numerical method's order of accuracy, which is  $\mathcal{O}(\Delta t, \Delta x)$  here. To show the latter, in Figure 7.13b the time

evolution of the relative mass error is given for a grid and time step twice as fine as those corresponding with Figure 7.13a. Both the oscillation's amplitude and the linear behavior's slope appear to have been halved approximately. Due to the still relatively coarse resolution of the water column (8 and 16 cells, respectively), the mass error for water (not given) does not yet show asymptotic first-order convergence behavior.

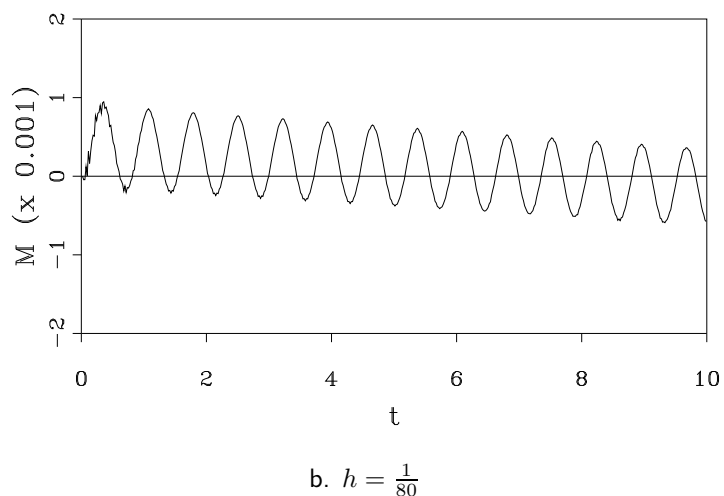
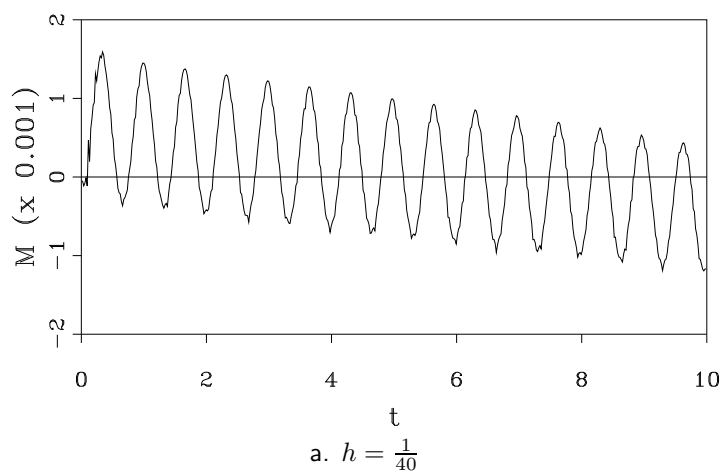


Figure 7.13: Time evolution of relative error in total mass of air in closed tube.

The present test case shows that the ghost-fluid method is not exactly conservative when pressure waves interact with two-fluid interfaces. Recently, Van Brummelen and Koren have proposed a two-fluid method which does not suffer from the pressure-oscillation problem and which is also exactly conservative [13]. Their method is applied to a test case in which a shock wave hits a water-air inter-

face. Neither zeroth-order pressure errors nor mass-conservation errors are observed in their numerical results.

## 7.5 Conclusions

To avoid large solution errors near interfaces (an intrinsic problem of capturing methods), four fixes have been proposed, three consisting of some locally non-conservative solution update and as the fourth a ghost-fluid fix. For density ratios of the order 1000 (typical water-air ratio) the non-conservative fixes fail, in the analysis as well as in the numerical experiments. As opposed to that, the ghost-fluid technique works. Even the computation of fronts running into vacuum ( $\frac{\rho_w}{\rho_a} = \infty$ ) is possible with the ghost-fluid method. Since in the ghost-fluid method only single-fluid fluxes are computed, it does not need a two-fluid Godunov approach. The first numerical computations performed with the discretization method are promising. Extensions to higher dimensions, higher accuracy, flows with gravity, etc. are challenging. The test case performed with the oscillating water column suggests that the method may lend itself particularly well for applications as, e.g., the impact of storm surges on ships and off- and onshore constructions. Compressibility plays an important role in these problems.

## CHAPTER 8

# Computation of Gravity-Driven, Immiscible Water-Air Flows

### 8.1 Introduction

In Chapter 6 we introduced a hyperbolic system which describes the dynamics of two immiscible, compressible fluids. This system represents the conservation of bulk mass and bulk momentum. An additional equation is added to describe the behaviour of the interface, separating the two fluids. The system of equations is balanced by specifying an equation of state relating the bulk density to the pressure. For this purpose Tait's equation of state is used. A further equation is added to establish the relation between the bulk density and the densities of the two separate species, for a given pressure. In the current chapter we compute solutions of this two-fluid system, under the action of gravity. We consider the flow of water underlying air (the normal situation in ship hydrodynamics), driven by the presence of the gravitational field. By considering both fluids to be compressible the gravity force enters the equations as a source term, resulting in a non-homogeneous hyperbolic system. In this chapter the effects of this non-homogeneity will be investigated, which will lead to an efficient way of implementing the source term into a numerical scheme. The spatial discretization of the system of equations is obtained using a finite-volume method with an approximate Riemann solver to determine the fluxes on the cell faces. Here the single-fluid linearized Godunov scheme, derived in Chapter 6, is used for this purpose. Furthermore the ghost-fluid method, see Section 7.3.4, is used to deal with the cells in which the interface is. All this will be demonstrated on a 1D model problem.

### 8.2 Equations of motion

#### 8.2.1 Fluid motion

We consider the flow equations (7.1a) for the bulk density and momentum, extended with the gravity force working in the negative  $x$ -direction. These equations read,

for an (open) interval  $\Omega \subset \mathbb{R}$ :

$$\int_{\Omega} \frac{dq}{dt} dx + f(q)_{\partial\Omega_{right}} - f(q)_{\partial\Omega_{left}} = Q \int_{\Omega} q dx, \quad \forall (x, t) \in \Omega \times \mathbb{R}^+, \quad (8.1)$$

where

$$q = \begin{pmatrix} \rho \\ \rho u \end{pmatrix}, \quad f(q) = \begin{pmatrix} \rho u \\ \rho u^2 + p \end{pmatrix}, \quad Q = \begin{pmatrix} 0 & 0 \\ -\text{Fr}^{-2} & 0 \end{pmatrix}. \quad (8.2)$$

$Q$  is the matrix due to the presence of the gravity force. Note that the source term is linear in the state vector  $q$ . The parameter  $\text{Fr} \equiv \frac{U}{\sqrt{gL}}$  is the Froude number, with  $U$  and  $L$  being a characteristic speed and length, and  $g$  the acceleration of gravity.

The system is again balanced by the bulk-density relation

$$\rho = \alpha(\phi)\rho_w(p) + (1 - \alpha(\phi))\rho_a(p), \quad (8.3)$$

in which  $\alpha$  is the volume-of-water fraction,  $\phi$  the level-set function and where  $\rho_w(p)$  and  $\rho_a(p)$  are the equations of state for water and air, respectively. The equation of state for both fluids has the form

$$\frac{p + Bp_{\infty}}{p_{\infty}(1 + B)} = \left( \frac{\rho}{\rho_{\infty}} \right)^{\gamma}, \quad (8.4)$$

where  $B, \rho_{\infty}$  and  $\gamma$  are constants depending on the type of fluid, and where  $p_{\infty}$  is the reference pressure for both fluids. The subscript  $\infty$  refers to some characteristic reference state. Excluding shock waves, the only discontinuity in the solution can occur across the interface.

### 8.2.2 Interface motion

To specify the location of the free surface, a level-set method is employed. The general idea of level-set methods is that an interface can be represented as a smooth hypersurface  $\Gamma(t) \in \mathbb{R}^n$  embedded in a smooth function  $\phi(\mathbf{x}, t) \in \mathbb{R}^{n+1}$  for which it holds

$$\Gamma(t) = \{\mathbf{x} \in \mathbb{R}^n, t > 0 : \phi(\mathbf{x}, t) = 0\}. \quad (8.5)$$

The motion of the interface  $\Gamma(t)$  follows that of the function  $\phi$ . Note that in our 1D model problem the interface  $\Gamma(t)$  is reduced to a point on the  $x$ -axis. First we will construct the initial value  $\phi(x, 0)$ . The straightforward way of constructing  $\phi(x, 0)$  is to let it be a signed-distance function. For a single interface, this is done through setting

$$\phi(x, 0) = \bar{x}(0) - x, \quad (8.6)$$

where  $\bar{x}(0)$  is the location of the free surface at  $t = 0$ . Note that  $\phi(x, 0)$  does not necessarily have to be a distance function, see, e.g., [44]. Secondly the time evolution of  $\phi(x, t)$ , and therefore also the interface location, is described by

$$\partial_t \phi + u(x, t) \partial_x \phi = 0, \quad \forall (x, t) \in \Omega \times \mathbb{R}^+, \quad (8.7)$$



with  $u(x, t)$  the underlying velocity field. This equation can be combined with the bulk-mass conservation equation to give

$$\partial_t(\rho\phi) + \partial_x(\rho u\phi) = 0, \quad \forall(x, t) \in \Omega \times \mathbb{R}^+. \quad (8.8)$$

Equation (8.8) can be incorporated into (8.1), which then has a state vector  $q(x, t) = (\rho, \rho u, \rho\phi)^T$  and flux function  $f(q) = (\rho u, \rho u^2 + p(\rho, \phi), \rho u\phi)^T$ . Note that  $p = p(\rho, \phi)$ . This dependence of  $p$  on  $\phi$  ( $\phi$  is not a passive scalar) is called feedback [44].

### 8.3 Analysis

In this section some analytical aspects of the 1D model problem will be established. Here we will consider the case where the level-set function  $\phi$  has not been incorporated into system (8.1). Incorporating the level-set function into the system is not expected to alter the analytical results which are obtained below. To accommodate the analysis we consider the system (8.1), written in quasi-linear differential form

$$\frac{\partial q}{\partial t} + A(q) \frac{\partial q}{\partial x} = Qq. \quad (8.9)$$

The system of equations is strictly hyperbolic as can be observed from the eigenvalues and eigenvectors of the flux Jacobian  $A(q) = f'(q)$ :

$$A(q) = \begin{pmatrix} 0 & 1 \\ c^2 - u^2 & 2u \end{pmatrix}. \quad (8.10)$$

The eigenvalues and (right) eigenvectors are collected in the matrices

$$\Lambda = \begin{pmatrix} u+c & 0 \\ 0 & u-c \end{pmatrix} \quad \text{and} \quad R = \begin{pmatrix} 1 & 1 \\ u+c & u-c \end{pmatrix}. \quad (8.11)$$

Incorporating the level-set equation would result in an additional eigenvalue  $\lambda = u$ .

After applying the transformation  $q = R w$ , with  $w : \mathbb{R} \times \mathbb{R}^+ \rightarrow \mathbb{R}^2$ , and after right multiplication with  $R^{-1}$ , system (8.9) can be rewritten as

$$\frac{\partial w}{\partial t} + \Lambda \frac{\partial w}{\partial x} = \tilde{Q} w, \quad \text{with} \quad \tilde{Q} = R^{-1} Q R = \frac{1}{2c\text{Fr}^2} \begin{pmatrix} -1 & -1 \\ 1 & 1 \end{pmatrix}. \quad (8.12)$$

The variables  $w$  are also known as the characteristic variables. Note that the system of equations is still coupled through the righthand side.

#### 8.3.1 Homogenization

In dealing with non-homogeneous systems of PDE's a number of different approaches has been developed. For instance in [39] it is shown that by incorporating the source term into the flux function, as opposed to evaluating the source term in

some straightforward manner, improvements in both accuracy and convergence rate are accomplished. This result justifies the search for transformations through which the present source term can be incorporated into the flux function as well. In the remainder of this section we will investigate a transformation which does precisely that.

In the hypothetical case that  $A(q) = 0$ , (8.9) reduces to a very simple differential equation: the linear ordinary differential equation

$$\frac{\partial q}{\partial t}(x, t) = Qq, \quad (8.13)$$

of which the solution reads

$$q(x, t) = e^{Qt}q(x, 0). \quad (8.14)$$

Given the fact that here  $Q$  is nilpotent of index 2 ( $Q^2 = 0$ ), (8.14) can be exactly expanded as

$$q(x, t) = (I + Qt)q(x, 0). \quad (8.15)$$

Given solution (8.15), for the realistic case  $A(q) \neq 0$  we postulate a solution of (8.9) of the form:

$$q(x, t) = (I + Qt)\tilde{q}(x, t). \quad (8.16)$$

(So, for  $A(q) = 0$ , it must hold:  $\tilde{q}(x, t) = q(x, 0)$ .) Substitution of (8.16) into (8.9) yields

$$\frac{\partial \tilde{q}}{\partial t} + B(q)\frac{\partial \tilde{q}}{\partial x} = 0, \quad \text{with } B(q) = (I + Qt)^{-1}A(q)(I + Qt). \quad (8.17)$$

Hence, equation (8.9) has been homogenized through the transformation (8.16). Substitution of the known matrices  $Q$  and  $A$  (see (8.2) and (8.10), respectively), yields for  $B(q)$ :

$$B(q) = \begin{pmatrix} -t\text{Fr}^{-2} & 1 \\ (c^2 - u^2) - t\text{Fr}^{-2}(2u + t\text{Fr}^{-2}) & 2u + t\text{Fr}^{-2} \end{pmatrix}. \quad (8.18)$$

The eigenvalue matrix and the (right) eigenvector matrix of  $B(q)$  are

$$\Lambda = \begin{pmatrix} u + c & 0 \\ 0 & u - c \end{pmatrix} \quad \text{and} \quad R = \begin{pmatrix} 1 & 1 \\ u + c + t\text{Fr}^{-2} & u - c + t\text{Fr}^{-2} \end{pmatrix}. \quad (8.19)$$

Note that the presence of the source term does not influence the eigenvalues of the Jacobian  $B(q)$  and that – as it should be – the non-homogeneous system (8.9) is still strictly hyperbolic because the eigenvalues of  $B(q)$  are real and distinct. Also note that the eigenvectors are now explicitly dependent on  $t$  and  $\text{Fr}$ . (Compare (8.19) with (8.11).) For sufficiently small  $t$ ,  $I + Qt \approx I$ , and thus (8.16) may be written as

$$q(x, t) \approx \tilde{q}(x, t), \quad (8.20)$$

and consequently (8.17) as

$$\frac{\partial q}{\partial t} + A(q) \frac{\partial q}{\partial x} \approx 0. \quad (8.21)$$

So, for sufficiently small  $t$ , an approximate solution method for (8.9) is to first find a solution  $q(x, t)$  of the known homogeneous hyperbolic system (8.21), and next to simply multiply this solution by  $I + Qt$ , i.e.,

$$q(x, t) := (I + Qt) q(x, t). \quad (8.22)$$

### 8.3.2 Steady-state solution property

It can be easily shown that the stationary solution of (8.1) must satisfy

$$H = \frac{1}{2}u^2 + \int \frac{dp}{\rho} + \text{Fr}^{-2}(x - \bar{x}) = \text{constant}, \quad (8.23)$$

where  $\bar{x}$  denotes the free-surface location. Relation (8.23) holds for every inviscid barotropic flow. The pressure integral can be evaluated with the use of (8.4) resulting in

$$\int \frac{dp}{\rho} = \frac{c^2}{\gamma - 1}, \quad (8.24)$$

where  $c$  is the speed of sound. Note that  $H$  is discontinuous at the free surface.

## 8.4 Numerical method

As mentioned before, the discretization of the level-set equation can be done independent of the Euler equations. This has the advantage that the level-set equation can be discretized differently from the system of Euler equations. Many discretizations have been explored for the Hamilton-Jacobi equation which the level-set equation in fact is, see [55]. In [55], Hamilton-Jacobi-type equations are solved by using techniques common for hyperbolic conservation laws. In the present work, equation (8.7) is discretized with the forward Euler method in time and with the first-order upwind, cell-centered finite-volume method in space. Numbering the finite volumes by  $i$  ( $i$  increasing in positive  $x$ -direction), and the discrete time level by  $n$ , the discrete version of (8.7) reads

$$\phi_i^{n+1} = \phi_i^n - \lambda \frac{u_i^n + |u_i^n|}{2} (\phi_i^n - \phi_{i-1}^n) - \lambda \frac{u_i^n - |u_i^n|}{2} (\phi_{i+1}^n - \phi_i^n), \quad (8.25)$$

where  $\lambda = \frac{\Delta t}{\Delta x}$ . The discretization of the system of Euler equations is also obtained through forward Euler and a first-order accurate, cell-centered finite-volume method. Omitting the source term, this discretization reads

$$q_i^{n+1} = q_i^n - \lambda (F(q_i^n, q_{i+1}^n) - F(q_{i-1}^n, q_i^n)), \quad (8.26)$$

where  $F(., .)$  is the numerical flux function, which is calculated using a single-fluid Godunov-type scheme.

The source term in the Euler equations is dealt with in the following manner. The property that for  $t \downarrow 0$  the solution of (8.9) converges to the solution of (8.21) inspired us to a two-stage solution algorithm for the non-homogeneous system, in which we do the following per time step:

- I. For a time step  $\Delta t$ , solve (8.1) with  $Q = 0$ . This can be done with one's own preferred time integrator.
- II. Multiply the solution  $q$  obtained from step 1 with  $I + \Delta t Q$ . Here the property that  $Q$  is nilpotent of index 2 is exploited.

This procedure may be categorized as a splitting method [67]. However, it is not a standard splitting method. Per time step, it requires the solution of a single initial-boundary value problem only, as opposed to the two initial-boundary-value problems that need to be solved in standard splitting methods.

The water-air interface is captured using the *ghost-fluid* method as described in Section 7.3.4.

## 8.5 Numerical results

The numerical test case considered in this chapter is the steady-state solution of (8.1) on the interval  $\Omega = \{x : x \in [0, 1]\}$  for the initial solution

$$u(x, 0) = 0, \quad p(x, 0) = 1, \quad \forall x \in \Omega, \quad (8.27)$$

with the free surface initially set at  $\bar{x}(0) = 0.5$ , and with  $\text{Fr} = 0.5$ ,  $\rho_w(x, 0) = 1$ ,  $x < 0.5$  and  $\rho_a(x, 0) = 10^{-3}$ ,  $x > 0.5$ . The boundary conditions imposed are  $u(0, t) = 0$  and  $p(1, t) = 1$ . So, at  $x = 0$  the tube is closed and at  $x = 1$  it is open. A situation sketch is given in Figure 8.1.

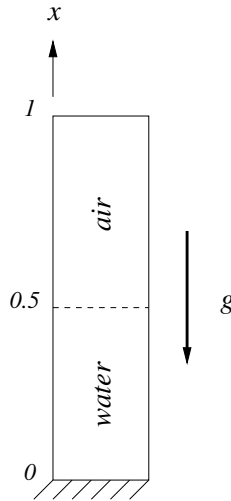


Figure 8.1: Air column on top of water column.

For the steady solution it holds  $u(x, t) = 0$ . A lower bound for the steady pressure at the bottom of the tube ( $x = 0$ ) is

$$p(x = 0) = p(x = 1) + \frac{(\rho_a)_{init}(l_a)_{init} + (\rho_w)_{init}(l_w)_{init}}{\text{Fr}^2}, \quad (8.28)$$

where  $(l_a)_{init}$  and  $(l_w)_{init}$  are the lengths of the initial air and water columns, respectively. An upper bound is

$$p(x = 0) = p(x = 1) + \frac{(\rho_a)_{init}2(l_a)_{init} + (\rho_w)_{init}(l_w)_{init}}{\text{Fr}^2}. \quad (8.29)$$

We impose the lower bound (8.28). For the initial solution chosen, (8.28) gives the exact bottom pressure value up to and including the third decimal digit:  $p(x = 0) = 3.002$ . Integration of the hydrostatic pressure equation

$$\frac{dp}{dx} = -\text{Fr}^{-2}\rho(p), \quad (8.30)$$

with  $\rho(p)$  given by (8.4), yields for water and air, respectively,

$$p_w(x) = \left( \frac{-(\rho_\infty)_w}{\text{Fr}^2(p_\infty(1+B))^{1/\gamma_w}} \frac{\gamma_w - 1}{\gamma_w} x + (p(x=0) + Bp_\infty)^{\frac{\gamma_w - 1}{\gamma_w}} \right)^{\frac{\gamma_w}{\gamma_w - 1}} - Bp_\infty, \quad (8.31)$$

$$p_a(x) = \left( \frac{-(\rho_\infty)_a}{\text{Fr}^2(p_\infty)^{1/\gamma_a}} \frac{\gamma_a - 1}{\gamma_a} (x - 1) + (p(x=1))^{\frac{\gamma_a - 1}{\gamma_a}} \right)^{\frac{\gamma_a}{\gamma_a - 1}}. \quad (8.32)$$

Equating both pressure distributions yields the location of the free surface  $\bar{x}$ . With the numerical data at hand, this gives:  $\bar{x}_{exact} = 0.49998$ . Knowing these exact velocity and pressure distributions, the solution errors can be calculated. The method as proposed in this paper is tested on three different grids, viz. with  $\Delta x = 1/40, 1/80$  and  $1/160$ . The distributions of the pressure and velocity are plotted in Figures 8.2 and 8.3. Figure 8.3 directly represents the error in the velocity distribution.

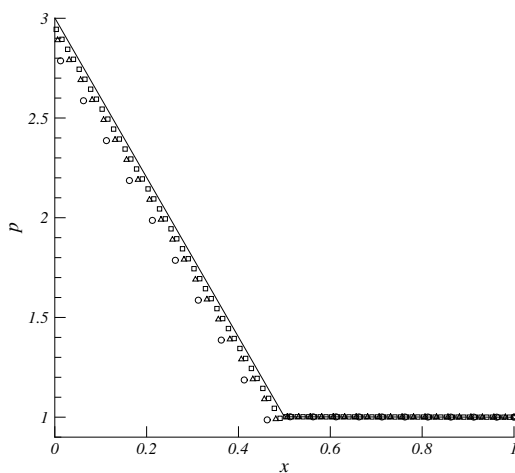


Figure 8.2: Pressure distribution for grids with  $\Delta x = \frac{1}{40}$  ( $\circ$ ),  $\Delta x = \frac{1}{80}$  ( $\Delta$ ) and  $\Delta x = \frac{1}{160}$  ( $\square$ ) and exact ( $-$ ). Only every second marker is shown.

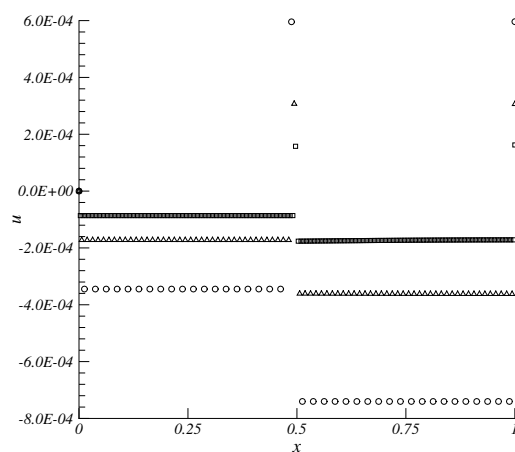
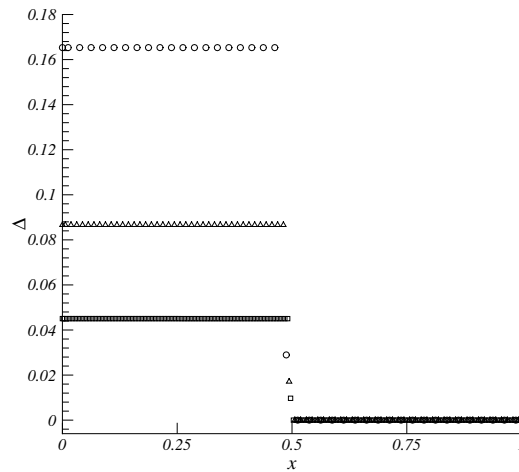


Figure 8.3: Velocity distribution for grids with  $\Delta x = \frac{1}{40}$  ( $\circ$ ),  $\Delta x = \frac{1}{80}$  ( $\Delta$ ) and  $\Delta x = \frac{1}{160}$  ( $\square$ ).

$(\Delta x)^{-1}$	$\frac{\Delta \bar{x}}{\Delta x}$
40	$-9.56 \times 10^{-3}$
80	$-12.32 \times 10^{-3}$
160	$-16.80 \times 10^{-3}$

Table 8.1: Relative error in free-surface location.

Figure 8.4: Pressure-error distribution for grids with  $\Delta x = \frac{1}{40}$  ( $\circ$ ),  $\Delta x = \frac{1}{80}$  ( $\Delta$ ) and  $\Delta x = \frac{1}{160}$  ( $\square$ ).

The distribution of the pressure error  $\Delta = |p_{num} - p_{exact}|$  is plotted in Figure 8.4. Inspection of the pressure-error distributions indicates that the pressure in pure water, and in the interface cell is resolved with first-order accuracy. Also notice the perfect first-order accuracy behaviour of the velocity. Even the solution wiggle in Figure 8.4 nicely converges to zero at an  $\mathcal{O}(h)$  rate. For each of the three numerical solutions, Table 8.1 still gives the relative error in the location of the free surface,  $\frac{\Delta \bar{x}}{\Delta x} = \frac{\bar{x}_{num} - \bar{x}_{exact}}{\Delta x}$ , with  $\bar{x}_{exact} = 0.49998$ . The free-surface location for the numerical solutions,  $\bar{x}_{num} = \bar{x}_{\phi=0}$ , has been determined by linear interpolation of the discrete values found. Its order of accuracy seems to be  $\frac{1}{2}$ .

The distribution of the total energy  $H$  according to (8.23) is given in Figure 8.5. Note the discontinuity at the free surface due to the fact that  $[c] \neq 0$  across the free surface.

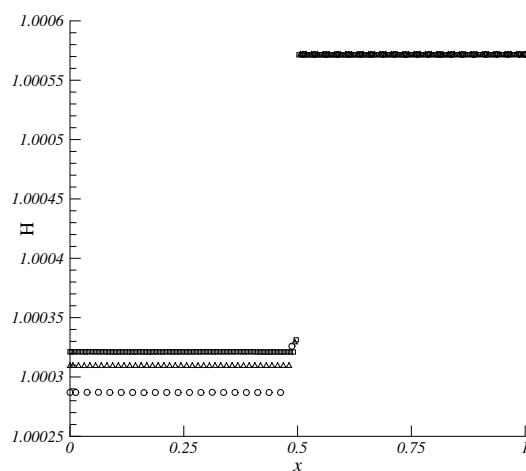


Figure 8.5: Distribution of total energy for grids with  $\Delta x = \frac{1}{40}$ , ( $\circ$ ),  $\Delta x = \frac{1}{80}$  ( $\Delta$ ) and  $\Delta x = \frac{1}{160}$  ( $\square$ ).

Finally, Figure 8.6 shows the distribution of the bulk density on the different grids. The bulk density is defined according to (8.3). These figures reveal that the large density jump is confined to a single cell.



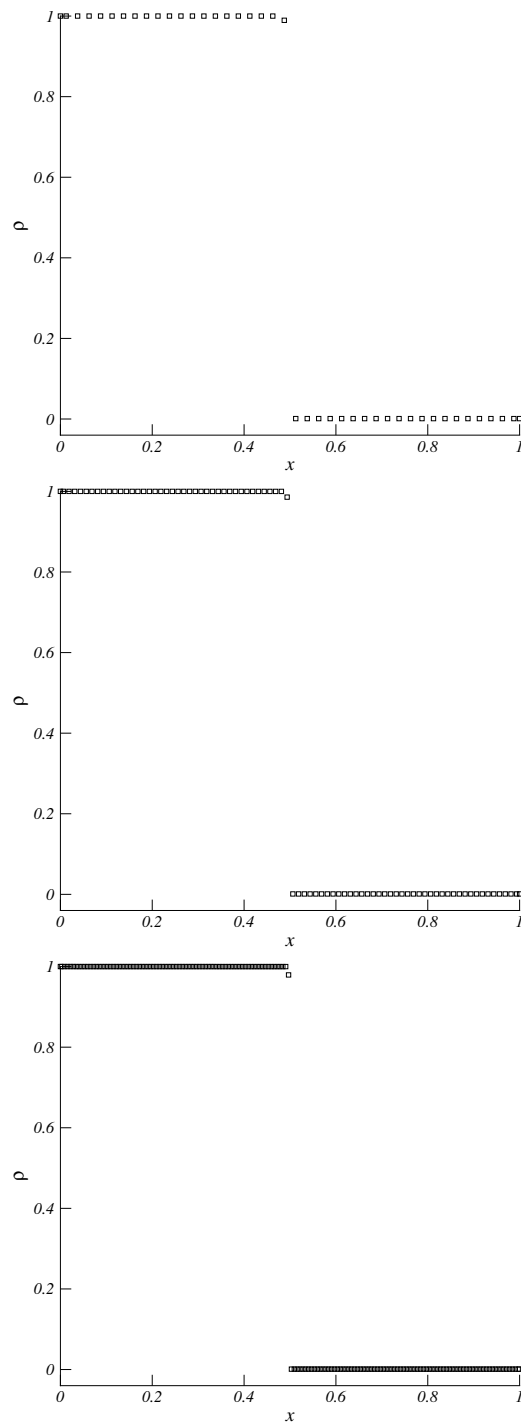


Figure 8.6: Bulk-density distributions for grids with  $\Delta x = \frac{1}{40}$ ,  $\frac{1}{80}$  and  $\frac{1}{160}$  (from top to bottom).

## 8.6 Conclusions

In this chapter we have computed the steady-state solution of a flow in a 1D tube, open at one side and containing two immiscible fluids, water and air. Most delicate feature in this problem is the water-air interface with its large density jump ( $\rho_w/\rho_a = 10^3$ ). Our computational method, built up out of numerical ingredients such as a level-set technique, a finite-volume discretization, a linearized Godunov scheme, a ghost-fluid approach and a splitting technique for the source term, appears to work fine. Applying first-order accurate state interpolation in the approximate Riemann solver, it yields first-order velocity and pressure accuracy in all finite volumes filled with water and also in the finite volume with the interface. Most important: our computational method does not smear the jump in the density; the large density jump is captured within one cell. Of possible interest for future research is the solution of the homogenized system of equations for large time (steps).

## CHAPTER 9

# Conclusions and Recommendations

The two parts of this thesis, *part A* and *part B*, describe the research done on a fitting method (*part A*) and a capturing method (*part B*) for the computation of free-surface flows.

### 9.1 Fitting

The subject of *part A* is the development of an efficient method for the computation of Navier-Stokes flow with a single free boundary, the free surface. This method is directed towards solving free-surface problems from the field of ship hydrodynamics. The method is the extension to  $3D$  of the  $2D$  non-monolithic free-surface algorithm described in [12]. At the heart of this free-surface algorithm lies the nonlinear quasi free-surface condition, which has been extended to  $3D$  in this thesis.

**Imposed pressure perturbation** To start, we computed the shape of the stationary free surface due to a pressure perturbation imposed on that surface. The free-surface algorithm appears to yield the proper 3D wave physics. A study of the convergence behaviour revealed that the free surface converges almost grid-independently. For linear and mildly nonlinear wave systems only a single free-surface update may be sufficient for finding the wave pattern to within engineering accuracy. An important result is the reduction of the inner iteration to a single work unit only (a single relaxation sweep). This reduction yields a major improvement in both robustness and efficiency. However, this improvement may be undone by reflections originating from the artificial lateral boundaries. The subject of non-reflecting boundary conditions for hyperbolic systems has received much attention and is well understood. In contrast, non-reflecting boundary conditions for steady, non-hyperbolic systems of equations (like the current) are not so well developed. Therefore, we simply introduced a far-field dissipation zone, which is aimed at preventing waves to reach the far-field boundaries. This approach works quite well for the problems considered here. The size of the dissipation zone and the stretching rate of the mesh size in the zone influence the convergence rate of the method. Finding the proper size and stretching rate requires some a-priori knowledge of the solution and experience of the code user. A more theoretical investigation into

the subject of non-reflecting boundary conditions, for this type of flow problem, would be useful. Numerical computations on a family of three grids (from very fine to twice and even four times coarser), using a second-order accurate discretization of the Navier-Stokes equations and the quasi free-surface condition, yields a free-surface wave of which the wavelength increases with decreasing mesh size. A Fourier-type dispersion analysis applied to the corresponding modified equations confirms this experimentally found wave elongation, not only qualitatively but even quantitatively.

**Piecewise-uniform mesh** We also investigated the suitability of a piecewise-uniform mesh in comparison with the standard exponentially stretched mesh, for our Navier-Stokes computations with free-surface water waves. An advantage of a piecewise-uniform mesh is its simple generation, which, in our opinion, made it an interesting candidate for implementation in a collective a-posteriori grid-adaptation method for boundary layers, wake, . . . and free surface. The results obtained indicate that a double-deck piecewise-uniform mesh performs comparably with the exponentially stretched mesh in terms of computational work. However, the piecewise-uniform mesh reveals a non-smooth solution behaviour at the interface between the coarse and fine mesh. This is a clear disadvantage, which may be fixed by moving the fine-coarse-grid interface to even more quiet solution regions. For the present ship-hydrodynamics work this may easily be too far away (not only outside the boundary layer and wake, but also outside the Kelvin wedge). A new triple deck piecewise-uniform mesh, suggested by a study of our Navier-Stokes truncation error, was still investigated. It was found that because of the presence of the extra fine-coarse-grid interface in an even less quiet part of the flow domain (the lower part of the boundary layer), here the solution accuracy is even less good than on the double-deck mesh.

**Series 60 hull** The free-surface algorithm has also been applied to a standard test case originating from ship hydrodynamics: the computation of the complete flow field, i.e., the shape of the free surface and the underlying Navier-Stokes flow, generated by a Series 60 hull at  $Fr = 0.316$  and  $Re = 10^6$ . Three solutions were computed here: (i) the solution belonging to a 'double-body' approximation, (ii) the solution belonging to a linearization of the free-surface flow, the 'uniform-flow' linearization and (iii) the solution of the fully nonlinear free-surface flow problem. In all three cases the method converged, but only in the latter two cases it yielded the proper wave physics. Our best numerical solution still lacks in amplitude, compared to the experimental results. The explanation is the first-order accurate discretization of the quasi free-surface condition that we had to apply for this test case. A detailed Fourier-type analysis showed that for finite Reynolds number, steady free-surface waves cannot exist for the reduced Navier-Stokes equations, in combination with the quasi free-surface condition, neither in the continuous case nor in the desired fully second-order accurate discrete case. Fortunately, the Fourier analysis also showed that for the reduced Navier-Stokes equations, steady waves can exist when discretizing the quasi free-surface condition at first-order accuracy (under the constraint that the mesh Reynolds number exceeds 2). The same Fourier analysis also suggests that to find second- or higher-order accurate, steady solutions, the full Navier-Stokes equations should be solved, instead of the reduced Navier-Stokes

equations. Our free-surface method does not suffer from the ‘contact-line’ problem, i.e., the incompatibility between the free-surface and boundary-layer conditions. The interaction between the boundary layer and the free surface is nicely visible in, e.g., extra local thickening of the boundary layer. The numerical results have shown that in the boundary-layer region a high-wave-number wave is present, which seems to be physically correct but of which the existence is not known from classical analytical theory on water waves. The robustness of the simple, efficient remeshing algorithm that we proposed for the free surface appeared to be too small for these short waves: they may induce mesh tangling. A fix in the context of free-surface fitting is to apply a robust grid generator in each free-surface update. A future alternative is the application of a free-surface capturing technique instead of a free-surface fitting technique.

**Fourier analysis** Fourier analysis appeared to be a very powerful tool for investigating the existence of steady free-surface waves and the dispersion properties of these waves in both the continuous and the discrete case. The smoothness of the current Navier-Stokes solutions and their wavy character is the explanation for this success. It is foreseen that Fourier analysis can be further exploited for the present ship-hydrodynamics computations, maybe directly in 3D.

**Newton iteration** It is important to note that the current free-surface algorithm transferred the computational complexity to the solution of sub-problems, i.e., stationary Navier-Stokes boundary-value problems with the quasi free-surface condition imposed on one part of its boundary. Here, Newton’s method was used to deal with the nonlinearity, and preconditioned GMRES to solve the linear systems. In the numerical studies performed in *part A* of this thesis, solving the linear systems did not seem to be the bottleneck. The main difficulty is the convergence of Newton’s method. Decreasing the tolerance for GMRES does not improve the nonlinear convergence. A more detailed investigation of the nonlinear iteration is required in order to increase the computational efficiency of the overall method.

## 9.2 Capturing

**Two-fluid Godunov-type scheme** Since free-surface flows can be considered as a special case of two-fluid flows, and since capturing methods for free surfaces hold out the promise of generality and robustness (no problems with breaking waves, free-surface bifurcations, etc.), in *part B* of this thesis we concentrated on capturing methods for barotropic, compressible two-fluid flows. We first presented a system of hyperbolic conservation laws for the bulk flows and a level-set equation to distinguish between the two fluids. A finite-volume discretization was applied and a two-fluid, linearized Godunov scheme was derived to compute the fluxes at the cell faces. This scheme is attractive because of its simplicity and computational efficiency in combination with its good physical properties.

**Pressure-oscillation problem** A difficulty intrinsic to capturing methods is the presence of a zeroth-order error in the pressure, at and near the two-fluid interface. Here, four fixes for this error have been investigated, three consisting of some locally advective solution update and as the fourth a simple ghost-fluid fix was proposed.

For density ratios of the order  $10^3$  (typical water-air ratios), the advective fixes fail, as opposed to the ghost-fluid fix. However, a slight disadvantage of the ghost-fluid method is that it is not exactly mass conserving. Fortunately, the mass-conservation error does not behave zeroth-order accurate; it decreases with decreasing mesh size.

**Gravity** The system of conservation laws was extended with a source term to account for the gravitational force. Since this source term is linear an elegant splitting technique could be used together with the numerical ingredients already introduced in the previous chapters of *part B*. The resulting method was used to compute the flow in a tube which is partially filled with water. The numerical solution showed a wiggle near the interface which converges nicely to zero at an  $\mathcal{O}(h)$  rate as all other errors do. The important result is that our method does not smear the jump in the density over a number of cells; the interface is captured within a single cell only.

**Perspective** Capturing (and tracking) methods for two-fluid and two-phase flows are very promising and developing rapidly. During the writing of this thesis, for compressible, barotropic two-fluid flows, a conservative and pressure-oscillation-free formulation was proposed by van Brummelen and Koren. At present, an extension of the latter formulation is made to non-barotropic equations. Simplification to incompressible flows are also foreseen. As far as ship hydrodynamics is concerned, the perspective is the capability to compute flows which are hard or impossible to handle with fitting methods (transom-stern flows, flows with breaking waves, ...).

## References

- [1] C. Aalburg. Experiments in minimizing numerical diffusion across a material boundary. Master's thesis, Department of Aerospace Engineering, University of Michigan, 1996. Also: <http://www.engin.umich.edu/research/cfd/publications/publications.html>.
- [2] R. Abgrall. How to prevent pressure oscillations in multicomponent flow calculations: a quasi conservative approach. *Journal of Computational Physics*, **125**: 150–160, 1996.
- [3] R. Abgrall and S. Karni. Computations of compressible multifluids. *Journal of Computational Physics*, **169**: 594–623, 2001.
- [4] D. Adalsteinsson and J.A. Sethian. The fast construction of extension velocities in level set methods. *Journal of Computational Physics*, **148**: 2–22, 1999.
- [5] B. Alessandrini and G. Delhommeau. Simulation of three-dimensional unsteady viscous free-surface flow around a ship model. *International Journal for Numerical Methods in Fluids*, **19**: 321–342, 1994.
- [6] B. Alessandrini and G. Delhommeau. A multigrid velocity-pressure free-surface-elevation fully coupled solver for calculation of turbulent incompressible flow around a hull. In *Proceedings of the 21<sup>th</sup> Symposium on Naval Hydrodynamics (Trondheim, 1996)*, 1996.
- [7] R. Aris. *Vectors, Tensors, and the Basic Equations of Fluid Mechanics*. Dover, New York, 1962.
- [8] T. Aslam. A level-set algorithm for tracking discontinuities in hyperbolic conservation laws I. Scalar equations. *Journal of Computational Physics*, **167**: 413–438, 2001.
- [9] G.K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, Cambridge, 1967.
- [10] F. Bet. *Numerische Simulation reibungsbehafteter Schiffsumströmungen mit freier Oberfläche*. PhD thesis, Gerhard-Mercator-Universität, Duisburg, 1999.
- [11] E.H. van Brummelen. Numerical solution of steady free-surface Navier-Stokes flow. Technical Report MAS-R0018, CWI, June 2000.

- 
- [12] E.H. van Brummelen. *Numerical Methods for Steady Viscous Free-Surface Flows*. CWI-Tracts, **134**. CWI, Amsterdam, 2003.
- [13] E.H. van Brummelen and B. Koren. A pressure-invariant conservative Godunov-type method for barotropic two-fluid flows. *Journal of Computational Physics*, **185**: 289–308, 2003.
- [14] E.H. van Brummelen, B. Koren, and H.C. Raven. Numerical solution of steady free-surface Navier-Stokes flow. In N. Satofuka, editor, *Proceedings of the First International Conference on Computational Fluid Dynamics (Kyoto, 2000)*, pages 305–320. Springer, Berlin, 2001.
- [15] E.H. van Brummelen, H.C. Raven, and B. Koren. Efficient numerical solution of steady free-surface Navier-Stokes flow. *Journal of Computational Physics*, **174**: 120–137, 2001.
- [16] C.O.E. Burg, K. Sreenivas, D.G. Hyams, and B. Mitchell. Unstructured non-linear free surface simulations for the fully-appended DTMB Model 5415 series hull including rotating propulsors. In *Proceedings of the 24<sup>th</sup> Symposium on Naval Hydrodynamics (Fukuoka, 2002)*, pages 171–186, 2002.
- [17] E. Campana, A. Di Mascio, P.G. Esposito, and F. Lalli. Viscous-inviscid coupling in free-surface ship flows. *International Journal for Numerical Methods in Fluids*, **21**: 699–722, 1995.
- [18] Y.C. Chang, T.Y. Hou, B. Merriman, and S. Osher. A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *Journal of Computational Physics*, **124**: 449–464, 1996.
- [19] S. Chen, B. Merriman, S. Osher, and P. Smereka. A simple level set method for solving Stefan problems. *Journal of Computational Physics*, **135**: 8–29, 1997.
- [20] J.-P. Cocchi and R. Saurel. A Riemann problem based method for the resolution of compressible multimaterial flows. *Journal of Computational Physics*, **137**: 265–298, 1997.
- [21] R. Courant and K.O. Friedrichs. *Supersonic Flow and Shock Waves*. Springer, Berlin, 1976.
- [22] S.F. Davis. An interface tracking method for hyperbolic systems of conservation laws. *Applied Numerical Mathematics*, **10**: 447–472, 1992.
- [23] J. Farmer, L. Martinelli, and A. Jameson. A fast multigrid method for the non-linear ship wave problem with a free surface. In W. Patel and F. Stern, editors, *Proceedings of the Sixth International Conference on Numerical Ship Hydrodynamics (Iowa, 1993)*, pages 155–172. National Academy Press, Washington, DC, 1993.



- 
- [24] P.A. Farrell, A.F. Hegarty, J.J.H. Miller, E. O’Riordan, and G.I. Shishkin. *Robust Computational Techniques for Boundary Layers*. Applied Mathematics **16**. Chapman & Hall/CRC, 2000.
- [25] R.P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*, **152**: 457–492, 1999.
- [26] S.K. Godunov. Finite difference method for numerical computations of discontinuous solutions of the equations of fluid dynamics (Cornell Aeronautical Laboratory translation from the Russian). *Mathematicheskii Sbornik*, **47**: 271–306, 1959.
- [27] F.H. Harlow and J.E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, **8**: 2182–2189, 1965.
- [28] B.T. Helenbrook, L. Martinelli, and C.K. Law. A numerical method for solving incompressible flow problems with a surface of discontinuity. *Journal of Computational Physics*, **148**: 366–396, 1999.
- [29] D.R. van den Heul, C. Vuik, and P. Wesseling. A staggered scheme for hyperbolic conservation laws applied to unsteady sheet cavitation. *Computing and Visualization in Science*, **2**: 63–68, 1999.
- [30] C.W. Hirt and B.D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, **39**: 201–225, 1981.
- [31] M. Hoekstra. *Numerical simulation of ship stern flows with a space-marching Navier-Stokes method*. PhD thesis, Delft University of Technology, 1999.
- [32] T.Y. Hou, Z. Li, S. Osher, and H. Zhao. A hybrid method for moving interface problems with application to the Hele-Shaw flow. *Journal of Computational Physics*, **134**: 236–252, 1997.
- [33] M. Israeli and S. Orszag. Approximation of radiation boundary conditions. *Journal of Computational Physics*, **41**: 115–135, 1981.
- [34] P. Jenny, B. Müller, and H. Thomann. Corrections of conservative Euler solvers for gas mixtures. *Journal of Computational Physics*, **132**: 91–107, 1997.
- [35] S. Karni. Multicomponent flow calculations by a consistent primitive algorithm. *Journal of Computational Physics*, **112**: 31–43, 1994.
- [36] S. Karni. Far-field filtering operators for suppression of reflections from artificial boundaries. *SIAM Journal on Numerical Analysis*, **33**: 1014–1047, 1996.
- [37] S. Karni. Hybrid multifluid algorithms. *SIAM Journal on Scientific Computing*, **17**: 1019–1039, 1996.

- [38] F.J. Kelecy and R.H. Pletcher. The development of a free surface capturing approach for multidimensional free surface flows in closed containers. *Journal of Computational Physics*, **138**: 939–980, 1997.
- [39] B. Koren. A robust upwind discretization method for advection, diffusion and source terms. In C.B. Vreugdenhil and B. Koren, editors, *Numerical Methods for Advection-Diffusion Problems, Notes on Numerical Fluid Mechanics*, **45**, pages 117–138. Vieweg, Braunschweig, 1993.
- [40] B. Koren and M.R. Lewis. Computation of compressible, immiscible water-air flows under the action of gravity. In *Proceedings of the ECCOMAS Computational Fluid Dynamics Conference 2001 (Swansea, 2001)*. Appeared on CDROM, 2001.
- [41] B. Koren, M.R. Lewis, E.H. van Brummelen, and B. van Leer. Riemann-problem and level-set approaches for homentropic two-fluid flow computations. *Journal of Computational Physics*, **181**: 654–674, 2002.
- [42] B. Koren, M.R. Lewis, E.H. van Brummelen, and B. van Leer. Riemann-problem and level-set approaches for two-fluid flow computations I. Linearized Godunov scheme. Technical Report MAS-R0112, CWI, August 2001.
- [43] B. Koren, M.R. Lewis, E.H. van Brummelen, and B. van Leer. Riemann-problem and level-set approaches for two-fluid flow computations II. Fixes for solution errors near interfaces. Technical Report MAS-R0113, CWI, August 2001.
- [44] B. Koren and A. Venis. A fed back level-set method for moving material-void interfaces. *Journal of Computational and Applied Mathematics*, **101**: 131–152, 1999.
- [45] H. Lamb. *Hydrodynamics*. Dover, New York, 1945.
- [46] B. van Leer. Towards the ultimate conservative difference scheme, V. A second-order sequel to Godunov's method. *Journal of Computational Physics*, **32**: 101–136, 1979. Also: *Journal of Computational Physics*, **135**: 229–248, 1997.
- [47] M.R. Lewis and B. Koren. Efficient computation of steady, 3D water-wave patterns, applications to hovercraft-type flows. Technical Report MAS-R0232, CWI, December 2002. (submitted for publication in the *International Journal for Numerical Methods in Fluids*).
- [48] M.R. Lewis and B. Koren. Efficient computation of steady, 3D water-wave patterns. In *Proceedings of the 16<sup>th</sup> AIAA Computational Fluid Dynamics Conference (Orlando, 2003)*. AIAA-paper 2003-3973, also appeared on CDROM, 2003.
- [49] M.R. Lewis, B. Koren, and H.C. Raven. Computation of 3D steady Navier-Stokes flow with free-surface gravity waves. In S. Armfield, P. Morgan, and K.

- Srinivas, editors, *Proceedings of the Second International Conference on Computational Fluid Dynamics (Sydney, 2002)*, pages 100–105. Springer, Berlin, 2003.
- [50] M.J. Lighthill. *Waves in Fluids*. Cambridge University Press, Cambridge, 1978.
- [51] G.H. Markstein. Chapter B: Theory of flame propagation. In G.H. Markstein, editor, *Non-Steady Flame Propagation*, pages 5–14. Pergamon Press, New York, 1964.
- [52] W. Mulder, S. Osher, and J.A. Sethian. Computing interface motion in compressible gas dynamics. *Journal of Computational Physics*, **100**: 209–228, 1992.
- [53] B.D. Nichols and C.W. Hirt. Calculating three-dimensional free surface flows in the vicinity of submerged and exposed structures. *Journal of Computational Physics*, **12**: 234–246, 1973.
- [54] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Applied Mathematical Sciences, **153**. Springer, Berlin, 2003.
- [55] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, **79**: 12–49, 1988.
- [56] S. Osher and F. Solomon. Upwind difference schemes for hyperbolic systems of conservation laws. *Mathematics of Computation*, **38**: 339–374, 1982.
- [57] S.J. Osher and G. Tryggvasson (editors). Special issue on computational methods for multiphase flows. *Journal of Computational Physics*, **169** (2), 2001.
- [58] S. Piperno, C. Farhat, and B. Larrouturou. Partitioned procedures for the transient solution of coupled aeroelastic problems. part I: model problem, theory and two-dimensional application. *Computational Methods in Applied Mechanical Engineering*, **124**: 79–112, 1995.
- [59] H. Raven and H. van Brummelen. A new approach to computing steady free-surface viscous flow problems. In *Proceedings of the 1<sup>st</sup> MARNET-CFD workshop (Barcelona, 1999)*, 1999.
- [60] H.C. Raven. *A solution method for the nonlinear ship wave resistance problem*. PhD thesis, Delft University of Technology, 1996.
- [61] H.C. Raven and B. Starke. Efficient methods to compute steady ship viscous flow with free surface. In *Proceedings of the 24<sup>th</sup> Symposium on Naval Hydrodynamics (Fukuoka, 2002)*, 2002.
- [62] Y. Ren and R.D. Russell. Moving mesh techniques based upon equidistribution, and their stability. *SIAM Journal on Scientific and Statistical Computing*, **13**: 1265–1286, 1992.

- [63] H.G. Roos, M. Stynes, and L. Tobiska. *Numerical Methods for Singularly Perturbed Differential Equations, Convection-Diffusion and Flow problems*. Springer Series in Computational Mathematics, **24**. Springer, Berlin, 1996.
- [64] H. Schlichting. *Boundary-Layer Theory*. McGraw-Hill, New York, 1979.
- [65] J.A. Sethian. *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Material Science*. Cambridge University Press, Cambridge, 1996.
- [66] J.J. Stoker. *Water Waves: the Mathematical Theory with Applications*. Pure and Applied Mathematics, **IV**. Interscience, New York, 1957.
- [67] G. Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, **5**: 506–517, 1968.
- [68] M. Sussman, A.S. Almgren, J.B. Bell, P. Collela, L.H. Howell, and M.L. Welcome. An adaptive level set approach for incompressible two-phase flows. *Journal of Computational Physics*, **148**: 81–124, 1999.
- [69] M. Sussman and E.G. Puckett. A coupled level set and Volume-of-Fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of Computational Physics*, **162**: 301–337, 2000.
- [70] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, **114**: 146–159, 1994.
- [71] P.G. Tait. *Voyage of H.M.S. Challenger*. Stationery Office, London, 1889.
- [72] J.F. Thompson, B.K. Soni, and N.P. Weatherill, editors. *Handbook of Grid Generation*. CRC Press, 1999.
- [73] L. Tingqui, J. Matusiak, and R. Lehtimaäki. Numerical simulations of viscous flows with free-surface around realistic hull forms with transom. *International Journal for Numerical Methods in Fluids*, **37**: 601–624, 2001.
- [74] Y. Toda, F. Stern, and J. Longo. Mean-flow measurements in the boundary layer and wake field of a Series 60  $c_b = .6$  ship model for Froude numbers .16 and .316. Technical Report 352, Iowa Institute of Hydraulic Research, August 1991.
- [75] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics, A Practical Introduction*. Springer, Berlin, 1997.
- [76] G.D. Tzabiras. A numerical investigation of 2D, steady free surface flows. *International Journal for Numerical Methods in Fluids*, **25**: 567–598, 1997.
- [77] A. van der Ploeg. *Preconditioning for sparse matrices with applications*. PhD thesis, Rijksuniversiteit Groningen, 1994.

- 
- [78] A.E.P. Veldman. *Boundary layer flow past a finite flat plate*. PhD thesis, Rijksuniversiteit Groningen, 1976.
- [79] P. Wesseling. *Principles of Computational Fluid Dynamics*. Springer Series in Computational Mathematics, **29**. Springer, Berlin, 2000.
- [80] G.B. Whitham. *Linear and Nonlinear Waves*. Wiley, Chichester, 1974.
- [81] D.C. Wyatt. Development and assessment of a nonlinear wave prediction methodology for surface vessels. *Journal of Ship Research*, **44**: 96–107, 2000.
- [82] P.A. Zegeling. *Moving-Grid Methods for Time-Dependent Partial Differential Equations*. CWI-Tracts, **94**. CWI, Amsterdam, 1993.
- [83] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, **127**: 179–195, 1996.
- [84] J. Zhu and J.A. Sethian. Projection methods coupled to level set interface techniques. *Journal of Computational Physics*, **102**: 128–138, 1992.



# Samenvatting

## Numerieke methoden voor waterstromingen met vrije-oppervlaktegolven

De numerieke simulatie van gas- en vloeistofstromingen die (gedeeltelijk) begrensd worden door vrije randen kent vele technische toepassingen, zoals bijvoorbeeld de omstroming van schepen en off-shore constructies, de uitstroming in de atmosfeer van een gasstraal uit een straal- of raketmotor, of de stroming van glas in een glasoven. De bruikbaarheid in het ontwerpproces vraagt van deze numerieke methoden een grote nauwkeurigheid en robuustheid tegen een aanvaardbare prijs.

In het eerste deel van dit proefschrift (*Deel A*, welke de hoofdstukken 2 tot en met 5 omvat) beschouwen we een numerieke methode die geschikt is voor 3D, stationaire vrije-randproblemen waarbij de vrije rand wordt doorsneden door een 'drijvend' lichaam (drijvend tussen aanhalingstekens omdat het lichaam niet kan bewegen in de vloeistof; het wordt als het ware vastgehouden). Het algoritme, dat gebaseerd is op ideeën en inzichten verkregen uit numerieke methoden voor vrije-rand potentiaalstromingen, is een iteratief proces waarbij een reeks stationaire, 3D Reynolds-gemiddelde Navier-Stokes randwaardeproblemen wordt opgelost, met de zogenaamde quasi vrije-randvoorwaarde opgelegd op het geschatte vrije oppervlak. Deze quasi vrije-randvoorwaarde is een combinatie van de kinematische en dynamische vrije-randvoorwaarden.

Deze numerieke methode wordt getest aan de hand van een aantal voor de scheepshydrodynamica relevante testgevallen. De numerieke resultaten laten zien dat de methode tegen zeer acceptabele kosten de correcte golfphysica voorspelt, zelfs voor zeer sterke niet-lineaire golfsystemen. De robuustheid van de methode wordt verder verbeterd door de frequentie waarmee het vrije oppervlak wordt aangepast te verhogen. Deze verbeterde methode wordt toegepast bij het simuleren van de Navier-Stokes stroming, inclusief het golvende wateroppervlak, rondom een Serie 60 romp. De eerste numerieke oplossingen vertonen al redelijk goede overeenkomsten met in sleeptankproeven verkregen meetresultaten.

Hoofdstuk 5 bevat een theoretische analyse, met behulp van Fourier-theorie, van het gelineariseerde, instationaire vrij-oppervlak randwaardeprobleem. Voor het stelsel semi-discrete vergelijkingen wordt onder andere een voorwaarde afgeleid voor het bestaan van stationaire zwaartekrachtsgolven in onze numerieke berekeningen. Ook wordt een nauwkeurige analyse gedaan van de numerieke dispersiefout in de oppervlaktegolven.

In *Deel B* (hoofdstukken 6, 7 en 8) van dit proefschrift onderzoeken we een *interface-capturing* methode. Bij verwaarlozing van de viscositeit wordt hierbij de stroming beschreven door een stelsel inhomogene, niet-lineaire, hyperbolische partiële differentiaalvergelijkingen. De inhomogeniteit (bronterm) wordt veroorzaakt door de aanwezigheid van het zwaartekrachtsveld. Aangezien de bronterm lineair is kan er gebruik gemaakt worden van een elegante splitsingsmethode. Het systeem van vergelijkingen wordt gediscretiseerd middels Godunovs methode waarbij er, ten behoeve van de efficiëntie, een benaderende Riemann-oplosser wordt gebruikt. De lokatie van het scheidingsvlak tussen de twee fluïda wordt beschreven met behulp van de 'level-set' methode en in de reken(cel) waarin het scheidingsvlak zich bevindt wordt de 'ghost-fluid' methode gehanteerd. Deze aanpak leidt tot bevredigende resultaten voor de beschouwde testgevallen. Het grote voordeel van deze methode is dat de grote dichtheidssprong niet wordt uitgesmeerd over meerdere cellen, maar wordt gevangen in één cel.



## Curriculum Vitae

The author was born on September 24<sup>th</sup> 1971 in Zwolle (The Netherlands). Following his secondary education he attended, in 1991, the Department of Aeronautical Engineering of the School of Engineering and Technology in Haarlem, from which he graduated in 1995 obtaining his B.Eng.-degree. He subsequently continued his studies at the Faculty of Aerospace Engineering at the Delft University of Technology where he graduated in 1999 with an MSc-degree, on a thesis in theoretical fluid dynamics. After graduation he worked for a short period as a researcher, performing compressible Navier-Stokes-flow computations on a next-generation spacecraft model. In September 1999, the author was offered a PhD-position at the Centre for Mathematics and Computer Science (CWI) in Amsterdam. There, he was involved in the development of a Navier-Stokes method for water flows around moving ships, of which the results are described in this thesis. The author is currently employed as a lecturer in aeronautical engineering at the Hogeschool 'INHOLLAND'.