VISIONS • SCIENCE • TECHNOLOGY • RESEARCH HIGHLIGHTS

98

Verification of a homogeneous mixture model for the free surface problem

Ville Viitanen

VTT SCIENCE 98

Verification of a homogeneous mixture model for the free surface problem

Ville Viitanen

VTT Technical Research Centre of Finland Ltd

Thesis submitted for examination for the degree of Master of Science in Technology
Espoo, January 15, 2015

ISBN 978-951-38-8315-7 (Soft back ed.)
ISBN 978-951-38-8316-4 (URL: http://www.vttresearch.com/impact/publications)

VTT Science 98

ISSN-L 2242-119X
ISSN 2242-119X (Print)
ISSN 2242-1203 (Online)

JULKAISIJA – UTGIVARE – PUBLISHER

Teknologian tutkimuskeskus VTT Oy
PL 1000 (Tekniikantie 4 A, Espoo)
02044 VTT
Puh. 020 722 111, faksi 020 722 7001

Teknologiska forskningscentralen VTT Ab
PB 1000 (Teknikvägen 4 A, Esbo)
FI-02044 VTT
Tfn +358 20 722 111, telefax +358 20 722 7001

VTT Technical Research Centre of Finland Ltd
P.O. Box 1000 (Tekniikantie 4 A, Espoo)
FI-02044 VTT, Finland
Tel. +358 20 722 111, fax +358 20 722 7001

Juvenes Print, Tampere 2015

Aalto University
School of Engineering
Degree Programme in Mechanical Engineering

ABSTRACT OF
MASTER'S THESIS

Author: Ville Viitanen

Title:
Verification of a homogeneous mixture model for the free surface problem

Date: January 15, 2015 **Pages:** 227

Major: Marine Engineering **Code:** Kul-24

Supervisor: Professor Pentti Kujala

Advisors: Professor Emeritus Jerzy Matusiak
Jussi Martio, M.Sc. (Tech.)

In this thesis, the applicability of the homogeneous mixture model of FINFLO for the free surface problem is studied. The free surface problem is fundamental in marine hydrodynamics, and a special case in two-phase flows. The work explores the basis of this type of modelling from mathematical and numerical viewpoints, and verifies the mixture model for the problem.

The mathematical background of the problem is presented, together with the nature of it from the perspective of marine hydrodynamics. The bulk flow equations are usually averaged conditionally such that the governing equations of the multiphase model are formally the same as in the case of single-phase flow. It can be shown that one additional equation suffices for the description of the segregated phases. Here, the convection equation of the void fraction is utilized. The void fraction equation is derived in conservative form based on the incompressibility constraint of the individual phases.

The convection of the void fraction corresponds to the so-called Riemann problem. This is studied thoroughly by developing a two-dimensional solver for the comparison of some well-known schemes for the spatial discretization of the convective quantity. This solver is applied to the convection of a discontinuous distribution of the void fraction. In addition, the so-called SUPERBEE limiter is implemented to the FINFLO code for the extrapolation of the convective void fraction.

The numerical solution of the Navier-Stokes equations for simulations of two-phase flows is covered comprehensively. The code YAFFA, developed at the Aalto University, has a modern VOF model implemented, and for this reason, it is here used as a reference code. The solution algorithms, the computation of the convective quantities, the pressure correction stages as well as the treatment of the segregated phases in both of the codes are discussed in detail. The two-phase flow over a submerged ground elevation is computed using the codes FINFLO and YAFFA, and the forming free surface wave is compared to the corresponding results found from the literature.

The aim of this thesis is to get acquainted with the nature of the problem in conjunction with the specific methodology used to solve such flows. This is done in order to understand the requirements and possible modifications needed for the model when we wish to accurately predict ship flow phenomena that are not solvable using the traditional free surface tracking strategies. This way, the verification of the mixture model of FINFLO is achieved.

Keywords: VOF, CFD, Two-phase flow, Hydrodynamics, Free surface, FINFLO, YAFFA, Numerical modelling, Convection, Interface capturing

Language: English

Aalto-yliopisto
Insinööritieteiden korkeakoulu
Konetekniikan koulutusohjelma

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä:	Ville Viitanen		
Työn nimi:	Homogeenisen seosmallin verifiointi vapaan nestepinnan ongelmaan		
Päiväys:	26. toukokuuta 2015	**Sivumäärä:**	227
Pääaine:	Meritekniikka	**Koodi:**	Kul-24
Valvoja:	Professori Pentti Kujala		
Ohjaajat:	Emeritusprofessori Jerzy Matusiak Diplomi-insinööri Jussi Martio		

Tässä työssä tutkitaan FINFLOn homogeenisen seosmallin soveltuvuutta vapaan nestepinnan ongelmaan. Vapaan nestepinnan ongelma on keskeinen laivahydrodynamiikassa, ja samalla monifaasivirtauksien erikoistapaus. Työssä perehdytään tällaisen mallinnuksen perusteisiin matemaattisessa ja numeerisessa mielessä, ja verifioidaan samalla seosmallia tälle ongelmalle.

Työssä esitetään ongelman matemaattinen tausta sekä sen luonne laivahydrodynamiikan kannalta. Virtausta kuvaavat yhtälöt yleensä keskiarvostetaan ehdollisesti se. käytettävän monifaasimallin perusyhtälöt ovat muodollisesti samat, kuin yksifaasisessakin tapauksessa. Voidaan osoittaa, että tässä tapauksessa erillisten faasien kuvaukseen riittää yksi lisäyhtälö, joksi työssä otetaan aukko-osuuden konvektioyhtälö. Aukko-osuusyhtälö johdetaan säilymismuodossa perustuen faasien kokoonpuristumattomuusoletukseen.

Mainittu lisäyhtälö vastaa luonteeltaan konvektioyhtälön ns. Riemann-probleemaa, ja tätä käsitellään perusteellisesti. Työssä kehitetään kaksidimensioinen ratkaisija, jolla vertaillaan tunnettuja menetelmiä konvektoituvan suureen paikkadiskretoinnille soveltamalla sitä epäjatkuvan aukko-osuusjakauman konvektioprobleemalle. Lisäksi implementoidaan FINFLOon ns. SUPERBEE-rajoitin konvektoituvan aukko-osuuden ekstrapolointiin.

Työssä käsitellään kattavasti Navier-Stokes -yhtälöiden numeerista ratkaisua kaksifaasivirtausimulointimenetelmien kannalta. Referenssikoodiksi otetaan Aalto-yliopistossa kehitetty YAFFA, johon nykyaikainen VOF-malli on implementoitu. Muiden muassa koodien ratkaisualgoritmi, konvektoituvien suureiden laskenta, painekorjausvaihe sekä erottuneiden faasien käsittely kuvataan perusteellisesti. FINFLO- ja YAFFA -koodeilla lasketaan kaksifaasivirtaus vedenalaisen kummun yli, ja syntynyttä aaltokuviota verrataan myös kirjallisuudesta löytyviin tuloksiin.

Työn ajatuksena on tutustua vapaan nestepinnan ongelman luonteeseen yhdessä tällaisen yleisemmän ratkaisutavan kanssa. Tavoitteena on ymmärtää mallille asetettavia vaatimuksia sekä sitä, millaisia modifikaatioita siihen tulisi tehdä, kun esim. pyritään ennustamaan tarkasti sellaisia laivavirtauksiin liittyviä ilmiöitä, joihin perinteiset pintaa seuraavat mallit eivät pysty. Tällä tavalla saatiin FINFLOn seosmallin verifiointi aikaiseksi.

Asiasanat:	VOF, CFD, kaksifaasivirtaus, hydrodynamiikka, vapaa nestepinta, FINFLO, YAFFA, numeerinen mallinnus, konvektio
Kieli:	Englanti

Acknowledgements

I am grateful to my instructor and colleague, Jussi Martio, for his constant encouragement and support throughout this project and my career at VTT, as well as numerous on- and off-topic discussions on practically all matters in CFD. Similarly, I greatly appreciate the contribution from all my other VTT colleagues, past and present, that have led to this Master's Thesis. Tuomas Sipilä and Antonio Sanchez-Caja have had the patience to assist me with the mixture model of FINFLO. Timo Kukkanen introduced me to the fascinating world of seakeeping, and Heikki Helasharju to the exciting area of ship model testing. Ilkka Saisto, as a former team leader, has made this all possible. Also the financial support from VTT is gratefully acknowledged.

Professor Emeritus Jerzy Matusiak has acted as a supervisor and later as an instructor to this thesis. Not only for this project, but equally during the over four years of my studies within marine technology, Jurek's indispensable guidance and teachings, as well as faith and patience with me has produced this thesis. Jurek also gave valuable feedback on the manuscript of this thesis, for which I am grateful.

Professor Timo Siikonen has been an unofficial advisor for this work. I have had the opportunity to have many enchanting discussions with him about several aspects of this thesis. Timo has been a great teacher in CFD, and almost all I have learned about the subject I owe to him. Without the 24 cr. of the CFD courses he has held, this thesis would not have been feasible. Timo also provided valuable comments on the manuscript of this thesis, for which I owe my sincere appreciation.

I owe great gratitude to Tommi Mikkola. Starting as an instructor and supervisor of my Bachelor's Thesis, he has been of invaluable support through

the years. He has contributed vastly to my understanding about CFD, and next to none my work could have been completed to the present extent without this. He also provided excellent criticism and valuable feedback on the manuscript of this thesis.

Satu Hänninen has helped me considerably in getting started with YAFFA, for which I am most grateful. She, in addition to Otto Puolakka and Teemu Manderbacka, have given me excellent advice throughout my studies.

I express my sincere gratitude to my supervisor Professor Pentti Kujala for his support and guidance.

I am obliged to Professor Michel Visonneau of ECN for first suggesting the idea of studying the convection equation of the void fraction.

I also express my gratefulness to Janne Esa, who has proofread the manuscript of this thesis, in addition to being a good friend during all these years.

I must present my sincere appreciation to my parents, Leena and Pekka, as well as Merja and Jouko, for enormous support and encouragement with all in my life. I am also thankful to the rest of my family and friends for having borne with me during this process.

Espoo, May 26, 2015

Ville Viitanen

Abbreviations

BICS	Blended interface capturing scheme
BC	Boundary condition
CBC	Convective boundedness criterion
CFD	Computational fluid dynamics
CG	Conjugate gradient
CICSAM	Compressive interface capturing scheme for arbitrary meshes
DDADI	Diagonally dominant alternating direction implicit
FICS	Fast interface capturing scheme
GDS	Gamma differencing scheme
IGDS	Inter-gamma differencing scheme
LS	Level set
MPI	Message-passing interface
MUSCL	Monotone upstream schemes for conservation laws
NV	Normalized variable
NVD	Normalized variable diagram
NVF	Normalized variable formulation
ODE	Ordinary differential equation
PDE	Partial differential equation
QUICK	Quadratic upstream interpolation for convective kinematics
RANS	Reynolds-averaged Navier-Stokes
SIMPLE	Semi-implicit method for pressure-linked equations
STACS	Switching technique for advection and capturing of surfaces
STOIC	Second- and third-order interpolation for convection

TV	Total variation
TVD	Total variation diminishing
VOF	Volume of fluid
YAFFA	Yet another fine flow analyser

List of symbols

Roman symbols

A	Coefficient matrix in structured grids
A	Coefficient matrix in unstructured grids; amplitude
$\tilde{A}$	Jacobian of the flux vector
c	Speed of sound
C	Davidson coefficient in the Rhie-Chow interpolation; a known value
C_f	Friction coefficient
C	Courant number, $u\Delta t/\Delta x$
$\mathscr{C}$	Commutator between fields
d	Distance; pressure gradients in the Rhie-Chow interpolation
D/Dt	Substantial derivative
e	Specific internal energy
E	Total internal energy; error
f^*	Numerical flux
f	Blending function
$\vec{f}$	Force vector
$\vec{F}$	Inviscid flux vector in the x-direction
$\vec{F}_v$	Viscous flux vector in the x-direction
$\mathscr{F}$	Analytical flux
Fn	Froude number, $u/\sqrt{gL}$
$\vec{g}$	Acceleration due to gravity, $\lvert\vec{g}\rvert = 9.81\ \mathsf{m/s^2}$
$\vec{G}$	Inviscid flux vector in the y-direction

$\vec{G}_v$	Viscous flux vector in the y-direction
$\mathscr{G}$	Gravity contribution
h	Distance; enthalpy
$\vec{H}$	Inviscid flux vector in the z-direction
$\vec{H}_v$	Viscous flux vector in the z-direction
k	Kinetic energy of turbulence, $\frac{1}{2}\overline{u_i' u_i'}$
$\mathscr{L}$	Differential operator
L	Reference length
$\dot{m}$	Mass flux
$\vec{n}$	Normal vector
N	Number of computational cells
$\mathcal{O}$	Order of magnitude
p	Thermodynamic pressure; term in the BICS
$\bar{p}$	Mean turbulent pressure
p'	Pressure correction; fluctuating pressure
$\hat{p}'$	Pressure correction
q	Heat flux
$\vec{Q}$	Vector of source terms
r	Gradient ratio
$\vec{R}$	Residual vector
R_w	Wave making resistance
Re	Reynolds number, $\rho u L/\mu$
S	Surface area of a face of a control volume
t	Time
$\bar{u}$	Convection velocity, $\bar{u} = \vec{V} \cdot \vec{n}$; mean turbulent velocity
u'	Velocity correction; fluctuating velocity
$\vec{U}$	Vector of unknown variables
U	General variable; velocity
$\vec{V}$	Velocity vector, $\vec{V} = (u, v, w)$
$\mathscr{V}$	Control volume
$\mathcal{V}$	Discrete control volume
V	Computational domain
$\dot{\mathcal{V}}$	Volumetric flux
x	Mass fraction; spatial coordinate
$\vec{x}$	Point in space, $\vec{x} = (x, y, z)$

Greek symbols

α	Void or volume fraction; shape function
α_p	Under-relaxation factor for pressure corrections
α_u	Under-relaxation factor for velocity corrections
$\tilde{\alpha}$	Normalized void or volume fraction
β	Coefficient in the GDS; limit value in the BICS; a parameter controlling the manipulation for outflow BC
γ	Angle between a cell face and the interface normal; parameter in the Taylor extrapolation of the convection velocity
Γ	Mass transfer
δ_{ij}	Kronecker delta function
Δ	Change
ε	Dissipation of the kinetic energy of turbulence
θ	Angle of the velocity vector
λ	Wave length
κ	Coefficient in the MUSCL scheme; curvature
μ	Coefficient of dynamic viscosity
μ_t	Coefficient of turbulent dynamic viscosity
μ_k	Coefficient of turbulent dynamic viscosity in the $k-\omega$ model
μ_ω	Coefficient of turbulent dynamic viscosity in the $k-\omega$ model
ν	Coefficient of kinematic viscosity
ρ	Density
$\hat{\rho}$	Distance weighed density
σ	Coefficient of surface tension
σ_k	Turbulent Schmidt number in the $k-\omega$ model
σ_ω	Turbulent Schmidt number in the $k-\omega$ model
τ_{ij}	Stress tensor
ϕ	Limiter function in TVD methods; general variable
χ	Conditional averaging operator
ω	Angular frequency
Ω_i	i:th sub-domain containing only one field

Subscripts and superscripts

a	Air
av	Average
b	Bump
f	Face; friction
fs	Free surface
g	Gas
i	Component; index of a computational cell in structured grids
$i+1/2$	Index of a face of a computational cell in structured grids
j	Component; index of a computational cell in structured grids; index of a cell face in structured grids
$j+1/2$	Index of a face of a computational cell in structured grids
k	Index of a computational cell; mixture component
l	Index of a computational cell in unstructured grids; left side of a face; liquid
lm	Face between l:th and m:th computational cell in unstructured grids
m	Index of a computational cell in unstructured grids; step in the Runge-Kutta integration
n	Time or iteration level; normal direction
N, S, E, W, T, B	Bands of the coefficient matrix
p	Parallel
P	Diagonal band of the coefficient matrix; computational cell
r	Right side of a face
s	Differencing scheme; tangential direction
sb	SUPERBEE
T	Transpose
va	Van Albada
w	Water
x	x-component; derivative with respect to x
y	y-component; derivative with respect to y
z	z-component; derivative with respect to z

$\tilde{}$	Normalized value; infinitesimal value
∞	Reference
$+$	Left or right side of a face; dimensionless value
$-$	Left or right side of a face
$'$	Correction
$*$	Temporary term

Contents

Chapter 1

Introduction

This thesis is an expedition to a special case in two-phase flows, namely the free surface problem between two immiscible and segregated fluids, and its prediction with the methodology of computational fluid dynamics (CFD) using the Volume-of-Fluid (VOF) approach. The free surface problem is fundamental in marine hydrodynamics, and essentially a single most distinct feature distinguishing the field in the discipline of fluid dynamics. Due to the complexity of the situation, in this work, the physics are limited to the pure interface between a gas and a liquid, excluding any time-dependent phenomena or the effects of surface tension in addition to any surface piercing solids. The aim is to get acquainted with the nature of the problem in conjunction with the specific methodology used to solve such flows.

At the same time, the work presents a verification to a particular approach of two-phase modelling that reclines to the VOF strategy. This is to say the aim is to show the applicability of a (more) general purpose two-phase CFD methodology to the problem at hand. The VOF abbreviation has some ambiguity in it, but the approach taken here is shown to effectively reduce to an incompressible mono-fluid algorithm with varying fluid properties. Such is currently one of the most widely used models in computational marine hydrodynamics for CFD analyses involving the free surface. A homogeneous mixture model has been previously implemented in the Navier-Stokes code Finflo (Finflo, 2013), and that development forms a large part of this work.

This work consists of a computation of a test case that involves the free surface with two existing flow solvers. The other code is the Navier-Stokes solver YAFFA (Mikkola, 2009). This code has the standard VOF model implemented, and for this reason it is used here as a reference code. As an outcome one hopes to obtain details concerning their current applicability and subsequent requirements for computational cases resembling the one at hand. This work also includes development of a two-dimensional convection solver to study the numerical solution of the indicator equation used to identify the variable fluid properties in space and time. That part is done in order to assess the current deficiencies in FINFLO with respect to the free surface problem, as well as to obtain guidance for future work.

This thesis is organized as follows. The present chapter provides the background and introduction to the problem, as well as the perspective of a naval architect for the motivation to solve it. Varying solution strategies for a Navier-Stokes solver exist for the free surface problem, and these are briefly described. The second chapter describes the governing equations used in this work, and the third chapter deals with their discretization using the control volume technique. The fourth chapter introduces the two cases that are computed, one with the developed convection solver and the other with the two existing codes. Finally, the fifth chapter summarizes the work and draws future perspectives.

1.1 Background

Mathematically, the problem can be dissected unto such that one seeks the description for physical behaviour for a specific *field*, rather than restraining within a type of fluid (Stewart and Wendroff, 1984). Two or more fields seize disjointed and mostly time-dependent sub-domains within the global area of interest, involving an equal amount of systems of partial differential field equations governing their local state. This is illustrated in Fig. 1.1. The vector $\vec{U}_i$ depicts the state variable of interest for the i:th field, and $\mathscr{L}$ a differential operator. Each system is valid only in respective subset Ω_i. Moreover, the domains are time-dependent, *i.e.* $\Omega_i = \Omega_i(t)$. It is of substantial difficulty to solve such systems. An explicit distinction and time evolution of the different sub-domains would be necessary because the system

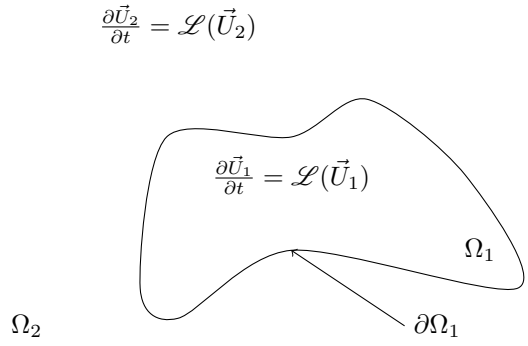


Figure 1.1: General two-field system.

requires boundary conditions, as well as complex interaction between the interfaces and the fields. In a discrete mesh, such would be thought to be attainable by the information from only a few computational cells.

One remedy to the above difficulty is the process of conditional averaging; it provides practical methods for prediction of multifield flow (Stewart and Wendroff, 1984). Such action can involve averaging by the space domain, time domain or more generally by ensembles (Hill, 1998). The application of a conditional averaging operator χ to the PDE describing the i:th field results in systems of the type

$$\frac{\partial \chi \vec{U}_i}{\partial t} = \mathscr{L}(\chi \vec{U}_i) + \mathscr{C}_i \ , \tag{1.1}$$

where $\mathscr{C}_i$ is a commutator. It denotes additional terms resulting from this averaging procedure which couples the different fields. This coupling would need either closure or modelling; a familiar analogue is the Reynolds stress tensor in resulting from Reynolds decompositioning and time averaging the Navier-Stokes equations, to filter the effects of turbulence. The situation does not differ in generic two-phase flows as the interface between two fluid systems requires description and details on the coupling between the phases. In this context some authors go into great detail within the derivation and modelling of the two-phase (two-fluid) systems in the sense of conditional averaging (cf. works of Hill (1998), Rusche (2002), Montazeri (2010); Montazeri et al. (2012, 2014), Hong and Walker (2000) and Weller (2005)). If one resides in a certain point $(\vec{x}, t)$ in space, generally all phases will be encountered in the measured quantity. Discrimination of whether the present

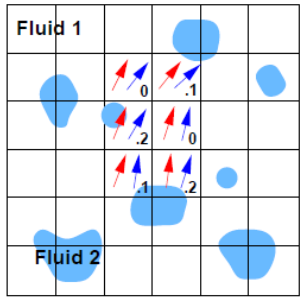


Figure 1.2: A sketch of a two-fluid model (Rusche, 2002).

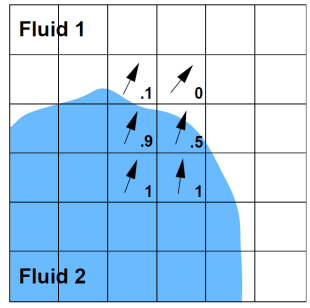


Figure 1.3: A sketch of the interface capturing model (Rusche, 2002).

phase is a gas (say), certain *conditioning* need to be imposed on the signal. This can be based on properties specific for that medium. The treatment is equivalent to multiplication of the quantity by an *indicator* that vanishes if the gas phase is not present at the point $(\vec{x}, t)$, and is unity otherwise (Hill, 1998, p. 331). This is to say an equation system is set up such that each fluid is treated as a distinct continuum, and such systems are applicable to all flow regimes since the topology of the flow is not prescribed (Rusche, 2002). This type of system is sketched in Fig. 1.2 denoting that each field is indeed an independent fluid.

The conditional averaging however results in additional terms in the balances of mass and momentum, all of which need additional modelling or closure in general multi-phase flows. This is due to the loss of information inherent in the averaging process. One way to simplify these is to impose different types of approximations, such as homogeneous, two-pressure or drift-flux models. These follow from different types of assumptions made within the derivation, and result in diverse systems of partial differential equations of the type of Eq. (1.1). (Hill, 1998; Rusche, 2002; Stewart and Wendroff, 1984)

Within the frame of this thesis, it is quite unnecessary to continue the previous discussion further. Rather, let us assume equal phasic velocities, casting the model to be of the homogeneous mixture type. Further, let us exploit the fact that the phases indeed are immiscible and completely segregated. This type of modelling is illustrated in Fig. 1.3. Additionally, we assume equal phasic pressures. The conservation equation for momentum reduces to that of the mixture whose properties are defined plainly through

an indicator function. Such function indicates the presence or absence of water, for instance, at a point in space and time. Hereafter, this indicator is equivalent to the phase void fraction, or volume fraction, α. Details of this derivation can be found from the work of Hill (1998). The conservation equations are formally the same as in the case of single-phase flow, and the commutator $\mathscr{C}_i$ between the fields vanishes. For computational purposes instead of two distinct flow areas only one is needed, for which the balance equations are applied. The two-phases are separated by an interface whose form implicitly follows from the flow solution, that is, the distribution of the void fraction function (probe). No boundary conditions are explicitly imposed on such a surface, neither one needs coupling terms for the phases since it is assumed that they do not interact in this sense. The mixture is defined to be either gas or liquid through its physical properties, such as density and viscosity, *i.e.* through the procedure of conditioning the signal. This provides a 'natural-like' sharp surface of contact discontinuity.

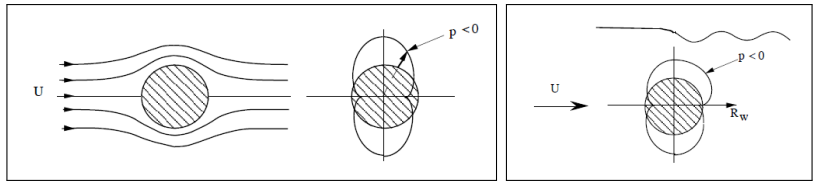
It is important to note, however, that the free surface problem *per sé* would not require a two-phase model, but a *surface* model. Nevertheless, it will be shown that the approach reduces in a sense to the standard VOF strategy, and one could argue that, for relatively simple shapes of the interface, no relevant flow details are suppressed within the model. It is worthwhile noting that such an approach leads to apparent difficulties in the vicinity of the interface where the material properties of the field change. By a quick inspection, such changes are discontinuous by nature. This poses a challenge to the algorithm used to solve the governing equations. The situation is in principle insensible as now it is the intention to model both phases as a single continuum. It means the interfaces can never be fully discontinuous within this approach. In addition, for the interpretation of the interface to be an actual boundary as the free surface between water and air, certain boundary conditions, namely the free surface boundary conditions, must be satisfied on the interface. Using the VOF solution strategy, the algorithm does not see a boundary between the fluids, but merely a varying distribution of material properties. The interface must somehow be interpreted from these distributions. It will be the topic of this thesis to determine whether the problem is solvable using such a model.

1.2 The free surface problem in naval hydrodynamics

Throughout the history of shipbuilding, the underlying principles have been the overall efficiency of the vehicle. Within hydrodynamics, this relies on considering all possible aspects in its motion such that the net resistance is resolved as well as the extent of current capabilities. It is of great importance to a shipyard to meet their proposed speed, the 'contract speed', with the installed engine power. Another equally important feature is the operability and seaworthiness of the vessel. That is, she is to survive the seaway expected on her path, and remain safe and stable during the intended operations. These define one of the design bases of ships. Failure in any regard cannot be considered an option and none should be included within the subtle art of compromise.

More specifically, a comprehensive description of a vessel afloat involves an exhaustive amount of detail from a variety of phenomena. Following Wackers et al. (2011) and including possible repercussions of interest here, one can list:

- **Ship resistance**. The air and water surrounding a vessel initially at rest impose aero and hydrodynamical forces as the ship tries to make its way through them. An accurate approximation of the machine power would require adequate description of both fluids and their interaction with the vessel.
- **Ship propulsion**. A propeller operating behind a ship changes the pressure field behind her. This in turn can affect the behaviour of the free surface at her aft, possibly causing ventilation.
- **Seakeeping**. As seaway composed of multiple different wave components encounter a ship, slamming type loads and high amplitude motions can occur. Their appropriate numerical prediction would require a method capable of determining a degree of 'merger' of the two fluids in a sense of non-linear free surface shapes.
- **Manoeuvring**. The incoming flow and wave field encounter the vessel obliquely, and complex flow separation near the free surface can occur.



(a) Infinite fluid depth (b) Finite fluid depth

Figure 1.4: Two-dimensional ideal flow past a cylinder (Matusiak, 2005).

A ship subtly starting to move will cause a disturbance to the surrounding fluid domains that are initially at rest. The situation can be idealized with Fig. 1.4. A two-dimensional flow of ideal fluid over a cylinder submerged in considerable depth causes symmetric pressure distribution around the object, and results in zero net resistance. The distribution changes when the cylinder is brought to a finite depth, resulting in a non-symmetric distribution of pressure. The presence of the free surface led to non-zero net resistance. Similarly, a ship acts as a moving disturbance, leading to deformation of the interface of air and water. Such is photographed in Fig. 1.5. Any water-borne vessel inevitably operates in the vicinity of the interface, and quite analogously to the cylinder the pressure distribution around the hull of the vehicle is altered by the free surface. The deforming free surface acts on the hull as normal and shearing stresses, and this gives rise to a term separated from the frictional drag, or the wave making resistance. The situation can analogously be seen as a dissipation of the kinetic energy of the ship to the formation of the wave field.

The advancing ship often initiates a wave system that involves breaking waves in its bow region (Fig. 1.6), include a surface piercing bulbous bow (Fig. 1.7), or involve different types of flow regions in its transom (Fig. 1.8). The deforming wave field in the stern region of the ship can interact with the propeller (Fig. 1.10). The seaway that the vessel encounters consists of waves of varying severity. Some may be steep enough such that slamming type loads will occur, or the motions of the vessel become significant to compromise comfort, or considerable green water incidents may take place. Harsh waters always pose risks to safe ship operation. A recent event of rough seas was reported in the Daily Mail[1], where an Antarctic cruise ship

[1] *"Pass the sea-sick pills: The terrifying moment a cruise ship was slammed by giant Antarctic waves"*, The Daily Mail, 10 December 2010

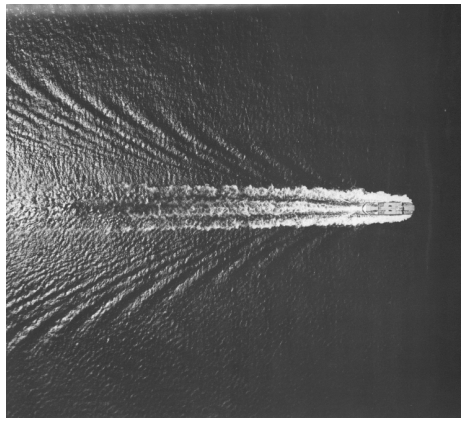


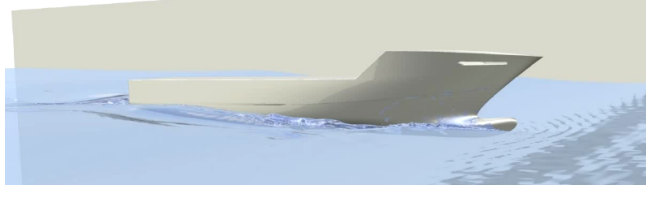
Figure 1.5: A wave system created by a moving ship. Photograph taken from van Dyke (1982).

encountered particularly sinister environmental conditions. A slamming bow wave was photographed during that voyage (Fig. 1.9). Appropriate capturing of phenomena such as these is essential to the design process.

In the above context, a naval architect is responsible for assuring that the vessel is efficient, economic, operable, safe and better than the design of the competing office. This is not a simple task, and involves expertise in many fields of hydrodynamics just to consider the above list alone. Practical design methods tend to move towards computer aided engineering due to the continuing increase in computer capabilities, and also computational fluid dynamics is getting a foothold in daily design of ships and marine structures. For example, the shape of the bulbous bow can be optimized such that minimum wave resistance is achieved at design speed (Fig. 1.11(a)), wave loads on offshore structures can be assessed (Fig. 1.11(b)) or the motions and stability of the vessel in six degrees of freedom in different circumstances can be determined (Fig. 1.11(c)). One interesting and quite recent development includes the air cavities to reduce the frictional surface of large ships 1.11(d)) where the approximations depend mainly on accurate computation of the wave in the 'trailing edge' of the cavity. Additional information on the phenomena involved in situations like those discussed above are detailed by Matusiak (2005), Hänninen (2014), Kukkanen (2012), Paulsen (2013), Larsson and Raven (2010), Stern et al. (2012), Eça et al. (2011), Bhushan et al. (2009), Wöckner-Kluwe (2013) and Shiri et al. (2014), for instance.

Figure 1.6: An example of wave breaking near the bow of a ship (Queutey and Visonneau, 2007).

(a) An *illustrative* example of surface piercing bulbous bow.



(b) Advancing ship with complex flow in the bow region.

Figure 1.7: Examples of surface piercing bulbous bows.

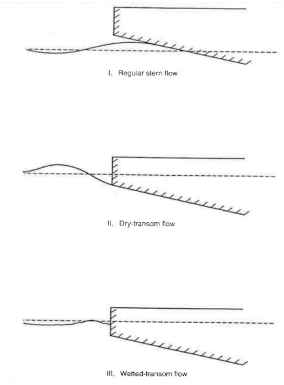


Figure 1.8: Possible flow regions in transom stern area (Larsson and Raven, 2010).

Figure 1.9: A slamming bow wave[1].

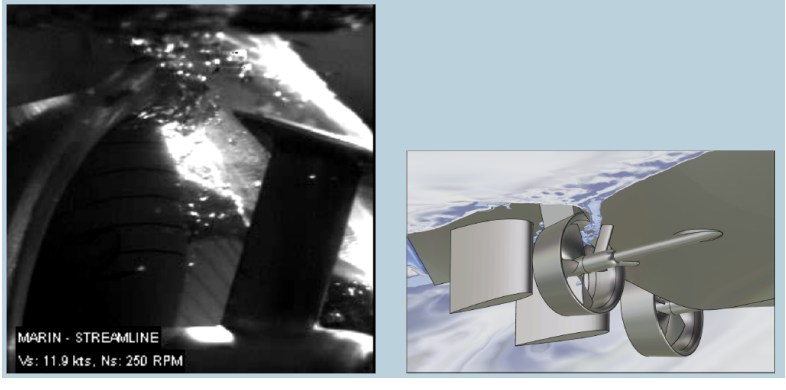
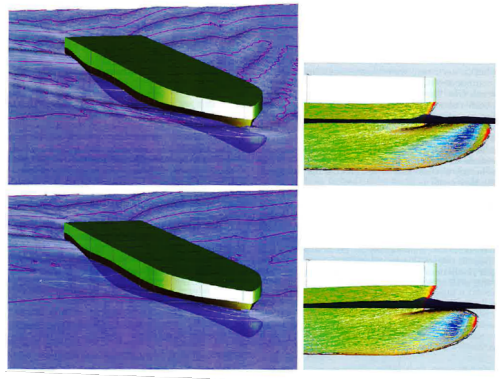
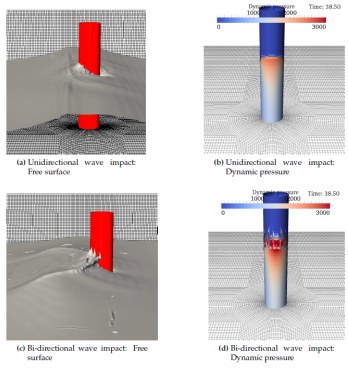


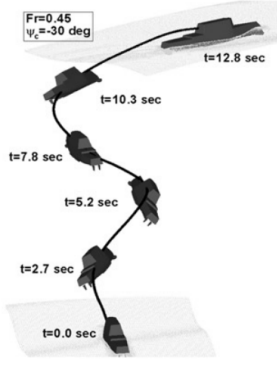
Figure 1.10: An example of interaction between the propeller and the free surface (Visonneau, 2013).



(a) An example of numerical prediction of waves generated by different shapes of bulbous bows (Larsson and Raven, 2010).



(b) An example of numerical predictions of wave loads on an offshore structure (Paulsen, 2013).



(c) An example of numerical predictions of vessel trajectory (Sadat-Hosseini et al., 2011).

(d) An example of air cavities to reduce the frictional drag.

Figure 1.11: Some illustrations of current capabilities of CFD predictions in naval architecture regarding the free surface problem.

1.3 Solution methods for the free surface problem

Flows with free surfaces include deforming boundaries that follow the bulk flow. They depend on the local flow solution; the local flow solution depends on the shape of the boundary. In the past, two clearly distinct methodologies have been developed for the numerical simulation of flows with free surfaces. They bear resemblance to the traditional solution strategies employed for flows with shocks. Without the loss of generality, these methods used are shortly summarized here. A classification can be made based on whether the interface is *tracked* or *captured* (Ferziger and Perić, 1999). For brevity, the focus is only on methodologies reclining on the CFD approach.

In **interface tracking**-type methodologies, the free surface shape is tracked. This is done by imposing the free surface boundary conditions *explicitly* on the actual free surface, that is, on one of the boundaries of the computational domain. This is to say the solution procedure actually 'sees' a boundary between the two fluids; thus in practice (of the kind that is relevant here) only the flow of the liquid phase needs to be considered and correspondingly the computational domain consists only of one constant property field in incompressible situations. The interface in this sense remains sharp throughout the solution, and the evolution is resolved very accurately, as the boundary of the computational domain moves with the interface. This is the consequence of the method itself. The kinematic boundary condition results directly into a partial differential equation for the free surface elevation. This equation is of the convection type, and can be integrated in time using similar routines that can be applied to the other field equations as well. The dynamic boundary condition is then satisfied directly on this surface. The tracking type methods can be further classified into subcategories based on the actual variable used in the solution, *e.g.* the height function or marker particles on the interface, or surface fitted techniques, as noted by Ubbink (1997). Applying and implementing methods based on the tracking of the free surface can be found in the works of Bet et al. (1996); Ferziger and Perić (1999); Meinander et al. (2006); Mikkola (1999, 2009); Muzaferija and Perić (1997); Schweighofer (2003); Wackers (2007); Wackers et al. (2011), for instance. An application of the interface tracking method in a ship flow case is shown in Fig. 1.12.

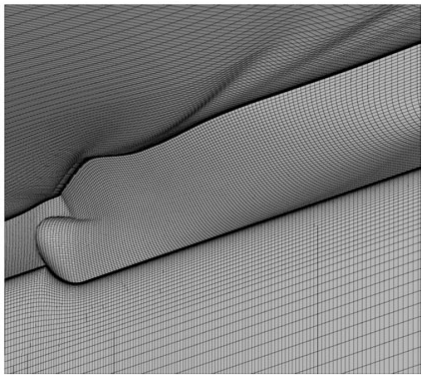


Figure 1.12: An illustration of the interface tracking approach. Figure taken from Wackers et al. (2011).

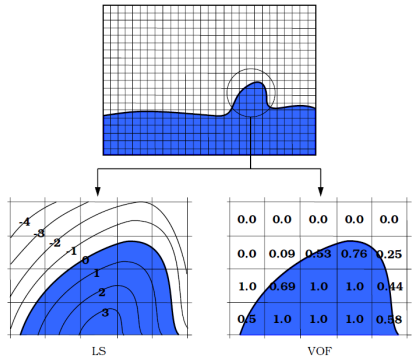


Figure 1.13: Illustration of the LS and VOF methods. Figure taken from Berberović (2010).

The **interface capturing** methodologies identify zones in the computational domain that are occupied by a specific fluid. That is, an indicator is used to keep track of the material properties, and interfaces between the fluids follow from these being different. For example, zones containing water are identified as such by cells in which the density is significantly larger, *i.e.*, that of water instead of air. This is a rather natural way of distinguishing the two (or more) fluids, and the volume fraction of a phase is convected along the bulk flow following the conservation of its mass. The solution procedure, however, does not see a boundary between the fluids, merely a distribution of their material properties; the interface itself follows from this distribution. The fact that the procedure is blind in this sense sets stringent requirements for the algorithm to keep the interface sharp, and special procedures must be taken to ensure this. Standard numerical schemes suffer from diffusive and unbounded behaviour. Another possibility is to use a level set function, which is a signed distance from the interface. That is, the zero-level contour represents the interface, and its time evolution follows from similar convection equation as does the volume fraction. This strategy is, however, completely artificial, namely the representation of the interface is mathematical; the level set function has no physical meaning in the form of a conservation law. The two commonly used strategies, the level-set (LS) and the VOF method, are depicted in Fig. 1.13.

Recent progress in the field of computational marine hydrodynamics also

shows an increased effort toward the use of a capturing model to account for the free surface. This is evident from the simple example of Fig. 1.14. A complex re-gridding procedure would be required if the simulated wave attains shapes deviating from the regular one, for instance. The capturing strategies form the only option for the majority of the problems introduced in the previous section. The treatment of the arbitrarily wetted transom stern is not a simple problem using the interface tracking method. A ship advancing in waves might cause diffractive waves with the oncoming waves such that the resulting wave field is no longer smooth, and the treatment of a partially dry bulbous bow is not possible. Furthermore, the estimation of propeller ventilation cannot be done with the interface tracking approach. None of these scenarios provide intrinsic difficulty to an interface capturing model such as the VOF approach, since the free surface can obtain arbitrary deformation. In addition, most commercial and non-commercial Navier-Stokes solvers tend to move towards the free surface capturing methods in ship flow cases (Hino, 2005; Larsson et al., 2000, 2010). For example, the ISIS-CFD (Queutey and Visonneau, 2007), OpenFOAM (Weller, 2005, 2008), FRESCO (Vaz et al., 2009), CFDSHIP-IOWA (Carrica et al., 2005) and FLUENT (Rhee et al., 2005), use the interface capturing approach as the *de facto* standard.

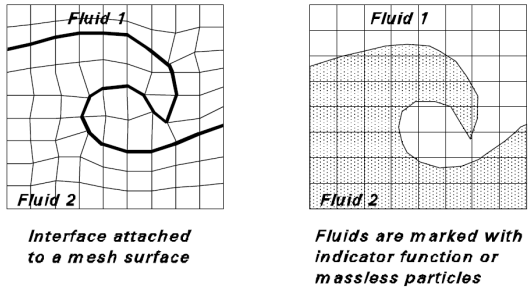


Figure 1.14: A difference between the interface tracking and interface capturing approaches. Figure taken from Ubbink (1997).

1.4 Scope of this work

The aim of this work is to focus on one specific numerical strategy to take the free surface into account in computational hydrodynamics. This is the

volume of fluid methodology to capture the interface between water and air. In ship flow situations, like those depicted in Figs. 1.6 – 1.11, the interface capturing strategies are practically the only viable option. It is mainly this fact that motivates the present work as well.

A homogeneous mixture model that utilizes the VOF methodology has been implemented in the Navier-Stokes solver FINFLO (Miettinen et al., 2006) currently in use at VTT Technical Research Centre of Finland. However, that development deals with flow situations that involve cavitating flows, mass transfer and wall boiling. A goal of this thesis is to verify the applicability of the homogeneous mixture model to the free surface problem, and identify the issues that need further attention. One should note that the free surface problem is essentially a special case in two-phase flows.

The two-phase VOF strategy consists of two main features with respect to pure single phase computations. First is the capability of the solution algorithm to handle the two different fields that are composed of vastly different properties and maintain the distinction of the zones occupied by a specific field. The second feature is the actual transport equation used to evolve the interface in time, at the same time maintaining its sharp resolving capabilities. Naturally, the two features are linked together. These issues will be detailed in the next chapters.

The former feature is approached with a study and application of the numerical algorithms used in both codes FINFLO and YAFFA. The latter code is used as a reference in order to verify that the solution procedure of FINFLO indeed falls into the same VOF-type strategy. The verification process is further facilitated with the computation of a two-phase flow case with simple geometry using both codes. The numerical uncertainty of the two-phase strategy is studied qualitatively. The time evolution of the void fraction is a distinct feature of the procedure and hence a computer program is developed to study this problem.

Chapter 2

Governing equations

In this chapter, the governing flow equations used in this study are presented. In addition, the basic concepts of the isothermic homogeneous mixture model and Volume of Fluid approach are described. The first section presents the governing flow equations based on the conservation of mass for the two separated fluids and conservation of momentum for the mixture. The second section describes the indicator equation used in capturing the interface between the two fluids. This equation falls within the traditional convection problem in CFD, which is briefly revised. In the third section, the free surface boundary conditions are discussed.

In this work the flow regions are considered as incompressible and immiscible with the two-phases separated. The phases are assumed to share the same velocity and pressure. The conservation of mass is formulated for both fluids, and the conservation of momentum is applied to the homogeneous mixture. The derivation of the flow equations, the continuity equation and the Navier-Stokes equations, can be found in the books of Tannehill et al. (1997) and White (2006), for example, for single phase flows and from the works of Stewart and Wendroff (1984), Weller (2005), Montazeri (2010), Hill (1998) and Rusche (2002) for more general multi-phase situations. The governing flow equations are merely recast without further proof in the next section. The transport equation for the volume fraction is derived in the subsequent section. It is here noted that both the standard form and the component form of the vector notation are used in this work, or $\vec{u}$ and u_i,

respectively. This should not cause confusion, and tensor procedures are used only when indicated.

2.1 Governing flow equations

In fluid dynamics, especially when considering flows within the naval realm, two fundamental laws of conservation are sufficient to be applied to the fluid media, in search of the pressure and velocity fields. These are conservation of mass and momentum, the latter being equivalent to Newton's Second Law of Motion. In naval hydrodynamics, it is just to allow for some simplifications to the description of the flow, and it is usually considered as isothermal and incompressible.

The conservation of mass in differential form for the fluid k without mass transfer is

$$\frac{\partial \alpha_k \rho_k}{\partial t} + \nabla \cdot \alpha_k \rho_k \vec{V} = 0 \ , \tag{2.1}$$

where α_k is the volume fraction for the k:th fluid, ρ_k its density and $\vec{V}$ is the velocity vector. The above equation simplifies in the incompressible limit ($\rho_{k,t} = 0$) to the convection equation of the void fraction (see Sec. 2.2). Eq. (2.1) is also referred to as the continuity equation. It merely states that, given the lack of sources or sinks for the mass, matter is neither created nor destroyed within this thesis. The conservation of momentum for the mixture is

$$\frac{\partial \rho \vec{V}}{\partial t} + \nabla \cdot \rho \vec{V} \vec{V} + \nabla p = \nabla \cdot \tau_{ij} + \rho \vec{g} \ , \tag{2.2}$$

where p is the pressure and τ_{ij} the components viscous stress tensor, and $\vec{g}$ the acceleration due to gravity. The first term on the left hand side denotes the time rate of change of momentum, and the second its convection. The components of the stress tensor are

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \ . \tag{2.3}$$

The clear majority of flows of interest to a naval architect are turbulent. The effects of turbulence on the equations above result in the Reynolds-averaged sense into an additional stress tensor, the Reynolds stress tensor. The velocity and pressure in turbulent situations are decomposed to their

mean and fluctuating parts, or

$$u_i = \bar{u}_i + u_i' \quad \text{and} \tag{2.4}$$

$$p = \bar{p} + p' \quad , \tag{2.5}$$

where i denotes any component ($i = x, y, z$) and the overbar denotes time average. When these are substituted into the Navier-Stokes equations, and the resulting equations averaged in time, one obtains a set of equations that governs the incompressible turbulent mean flow. The formulation results in additional terms, or to the so-called Reynolds stress tensor, the components of which can be combined with the stress tensor

$$\bar{\tau}_{ij} = \mu \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) - \rho \overline{u_i' u_j'} \ . \tag{2.6}$$

For the sake of completeness, the Reynolds-averaged Navier-Stokes (RANS) equations are then

$$\frac{\partial \rho \bar{u}_i}{\partial t} + \frac{\partial \rho \bar{u}_i \bar{u}_j}{\partial x_j} + \frac{\partial \bar{p}}{\partial x_i} = \frac{\partial \bar{\tau}_{ij}}{\partial x_j} + \rho g_i \ , \tag{2.7}$$

where the tensor notation is used. Henceforth the overbar is dropped for time-averaged variables. This thesis excludes the effects on turbulence in analyses that will follow. For numerical simulations, the standard SST $k-\omega$ is used to model the effects of turbulence on the flow. The reason for this exclusion, as well as some aspects of the effects from turbulence to the free surface problems, are discussed in Chap. 5.

Note that the conditional averaging required for decreasing the two fluid system into a single field representation is achieved through the time averaging of the momentum equations to filter the effects of turbulence and the conditioning of the conservation of mass to account for the two separated phases. If the conditioning procedure was done for the conservation of individual momenta, so-called inter-phase momentum transfer terms would appear in Eq. (2.2) as shown by Rusche (2002), for example. Further complexity is naturally encountered through the turbulence closures. The governing equations are formally the same as in the case of single phase flow. The approach taken here treats the fluid properties as variables across the

interface separating them. These are defined as

$$\phi = \sum_k \alpha_k \phi_k \ , \tag{2.8}$$

where ϕ is any fluid property, *e.g.* density. Two fluids accompanying the same computational cell share the same velocity and pressure. The fact that they are treated as a mixture requires the time evolution of the void fraction α that is convected with the flow velocity. This follows implicitly the solution of the bulk flow equations and is based on the conservation of individual phase mass. This discussion is resumed in the next section.

2.2 The volume fraction equation

One approach used to account for the presence of two segregated fluids is the use of a *mono-fluid* approach, *i.e.*, reducing the problem effectively to a single field computation. This enables the use of the single-phase equations in the solution procedure. The interface between the fluids follows implicitly from the solution of the bulk flow field, and manifests itself in the sharp transition of the mixture properties. The properties of the mixture consisting of k fluids are defined by

$$\rho = \sum_k \alpha_k \rho_k \quad \text{and} \quad \mu = \sum_k \alpha_k \mu_k \ , \tag{2.9}$$

where α is the indicator function denoting whether a flow region is filled with fluid k. The indicator is equivalent to the volume fraction, and it is a step function

$$\alpha_k(x, y, z, t) = \begin{cases} 1 & \text{if the point } (x, y, z, t) \text{ is in fluid } k \\ 0 & \text{if the point } (x, y, z, t) \text{ is not in fluid } k \end{cases} \tag{2.10}$$

presenting the probability that the phase k is present at a certain point in time and space. It is immediately visible that the fluid properties become discontinuous as well. This defines the interface as a contact discontinuity. With the definition of changing density in the fluid domain, it is clear that $\rho = \rho(x, y, z, t)$ in an Eulerian mesh. However, as both fluids are treated as incompressible, we have

$$\frac{D\rho}{Dt} = 0 \tag{2.11}$$

for a fluid particle. Now noting for simplicity that we are dealing with two fluids, there is only need for a single volume fraction via

$$\sum_k \alpha_k = 1 \ , \tag{2.12}$$

that is,

$$\alpha_1 = 1 - \alpha_2 = \alpha \ . \tag{2.13}$$

Settling to air and water, for example, the incompressible limit can be expanded

$$\frac{D}{Dt}\left(\alpha\rho_a - \alpha\rho_w + \rho_w\right) = 0 \ , \tag{2.14}$$

ρ_a is the density of air and ρ_w that of water. The above defines α as a Lagrangian invariant, or sets up an Eulerian mesh within which its value is defined by the movement of particles of water and air through it. For by writing an intermediate step from Eq. (2.14)

$$(\rho_a - \rho_w)\frac{D\alpha}{Dt} + \alpha\left(\frac{D\rho_a}{Dt} - \frac{D\rho_w}{Dt}\right) + \frac{D\rho_w}{Dt} = 0 \tag{2.15}$$

and following the previous assumptions, we end up with

$$\frac{D\alpha}{Dt} = 0 \ . \tag{2.16}$$

Noting that $\alpha = \alpha(x, y, z, t)$, and expanding between points (x_1, y_1, z_1, t_1) and (x_2, y_2, z_2, t_2) using a Taylor series

$$\alpha(x_2, y_2, z_2, t_2) \approx \alpha(x_1, y_1, z_1, t_1) + \frac{\partial\alpha}{\partial x_i}(x_{i,2} - x_{i,1}) + \frac{\partial\alpha}{\partial t}(t_2 - t_1) + \mathsf{H.O.T.} \tag{2.17}$$

and taking the limit of $t_2 \to t_1$, and accepting only the terms greater than $\mathcal{O}(\Delta x_i^2)$

$$\lim_{t_2 \to t_1} \frac{\alpha(x_2, y_2, z_2, t_2) - \alpha(x_1, y_1, z_1, t_1)}{t_2 - t_1} = \lim_{t_2 \to t_1} \frac{\partial\alpha}{\partial x_i}\frac{x_{i,2} - x_{i,1}}{t_2 - t_1} + \frac{\partial\alpha}{\partial t} \ , \tag{2.18}$$

where the summation notation over the repetitive index i was used for brevity. The total change, or the substantial derivative, of the volume frac-

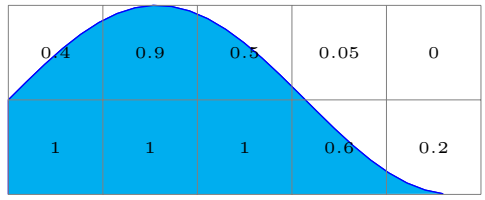


Figure 2.1: A schematic depiction of the VOF approach to capture the interface.

tion can now be written with its partial derivatives

$$\frac{D\alpha}{Dt} = \frac{\partial \alpha}{\partial t} + u_i \frac{\partial \alpha}{\partial x_i} \ . \tag{2.19}$$

As is seen, the interface evolution is based on the conservation of mass, and in this case follows directly from the assumption of incompressibility of the individual phases. It also implies that the kinematic boundary condition on the free surface is satisfied (Ferziger and Perić, 1999). The free surface boundary conditions are discussed in Section 2.3.

It was shown above that the value of α does not change throughout the domain. This is to say an initial quantity of the fraction is preserved throughout the space, and it is neither created nor destroyed. In yet other words, the volume fraction α is conserved within the solution domain. Eq. (2.19) is however in a primitive form. Albeit the relation is exact, the conservation of α in a discretized space consisting of finite computational molecules cannot be guaranteed. It would require an infinitesimal shrinking of the discretized domain, which is not possible for practical flow calculations. Luckily with the assumption of incompressibility it is possible, and more convenient for a numerical solution, to write Eq. (2.19) in a conservative form

$$\frac{\partial \alpha}{\partial t} + \frac{\partial \alpha u_i}{\partial x_i} = 0 \tag{2.20}$$

which, once discretized, is appropriate for solving the time evolution of the volume fraction. This equation essentially relates the time evolution of the interface to the local flow, and forms the basis of traditional VOF strategies. As one merely computes the values of the void fraction, complex re-gridding-like procedures are avoided (cf. Sec. 1.3) and the shape of the free surface can be determined from the distribution of α. This determination is illustrated

in Fig. 2.1. The location of the interface is usually relying on the iso-surface of $\alpha = 0.5$. Upon modelling the flow field with the stated equations (2.1 – 2.2) the only distinction between the two media follows from their material properties.

It is interesting that the governing equations are formulated for a continuum that now possesses a discontinuity. Clearly, the *exact* location of the free surface cannot be found from this distribution alone. There are means to construct the VOF-algorithm such that the interface will be sharp, that is, it is located between only a few cells. For the approach taken here, this requires appropriate discretization for the convection Eq. (2.20), as this alone evolves the void fraction in time, as well as possible modifications to the bulk flow solver. These will be introduced later. Eq. (2.20) falls in fact in the general category of a convection problem in fluid dynamics. Despite its deceivingly simple form the convection problem is an extremely difficult one to solve numerically when it involves a discontinuity in the dependent variable. One should note that based on the definition (2.10), we are dealing with such a problem. It has been a topic of intensive scientific effort since the 1970's, mainly initiated by the possibility of shock formation in gas dynamics since the Euler equations form a similar system. Those problems are briefly introduced below. Now Eq. (2.20) adds to the governing flow equations. The numerical solution of the convection equation of the void fraction and its influence in the discretized flow equations is addressed in Chap. 3.

2.2.1 The problem of convection

The brief inspection below serves as an introduction to the basic difficulties encountered with the numerical solution to the pure linear convection equation in the presence of a discontinuity in the dependent variable. It has been the matter of rigorous research in the past with many articles devoted to the numerical solution of the convection equation. Throughout the history of CFD, methods specifically developed for problems with discontinuities have been devised; cf. the works of Darwish and Moukalled (2006); Davis (1994); Ferziger and Perić (1999); Fletcher (1991); Gao et al. (2012); Harten (1983); Jasak (1996); Jasak and Weller (1995); Jasak et al. (1999); Leonard (1991); LeVeque (1992); Rusche (2002); Siikonen (2008); Sweby (1984); Tannehill et al. (1997); Ubbink (1997); Waterson and Deconinck (2007); Weller

(2005, 2008); Yee (1989); Yu et al. (2001), for instance. The time evolution of the void fraction is basically a Riemann problem. That is simply a conservation law together with initial data such that two constant states are separated by a single discontinuity (LeVeque, 1992). The problem is recast in one-dimensional form as

$$\frac{\partial \alpha}{\partial t} + \frac{\partial \alpha u}{\partial x} = 0 \tag{2.21}$$

with non-smooth initial data. With this single equation, the time evolution of α seems deceivingly simple. It is purely of convective nature. In addition, the pure convection problem has an exact transient solution over time Δt

$$\alpha(x, \Delta t) = \alpha_0(x_0 - u\Delta t) \ , \tag{2.22}$$

where α_0 is the initial distribution, x_0 its location at $t = 0$ and u a constant convection velocity.

It is the partial derivatives and their approximate (numerical) equivalents that make the convection equation of discontinuous quantity one of the most challenging problems in CFD. It is indeed the discontinuous variable that is of great interest within the VOF method, cf. Sec. 2.2. In principle we are dealing with the weak solution of the convection equation, since the derivatives are not defined at a discontinuity. It is not a solution of a partial differential equation in the classical sense[1]. This means that additional information needs to be provided for the approximative solution if one is to declare it a physically unique solution (Jasak, 1996; LeVeque, 1992; Sweby, 1984)[2]. The integral form of the equations continues to be valid over discontinuities in the dependent variables, but only if they satisfy the so called Rankine-Hugoniot jump conditions at the discontinuity (LeVeque, 1992; Siikonen, 2008).

Still, one might want to approximate Eq. (2.21) somehow to devise an algorithm to solve the convection problem numerically. The difficulty arises from the discontinuous nature of the discrete problem as well. There are two

[1] The term 'solution in the classical sense', or strong solution, denote that the dependent variable is n times continuously differentiable, with n being an integer

[2] In fact, recently a proof was submitted for the yet-to-be-found existence of smooth solutions for the full Navier-Stokes equations which is a Clay Institute millennium prize problem! Cf. http://www.claymath.org/sites/default/files/navierstokes.pdf for the problem formulation and http://www.math.kz/images/journal/2013-4/Otelbaev_N-S_21_12_2013.pdf for the latest attempted solution.

main causes for unsatisfactory numerical approximations of these contact discontinuities, and these are numerical diffusion and dispersion. These are well-known issues in numerical analyses, and their origin will be given below. Next, let us illustrate the difficulties with a few examples. Recalling the Taylor series expansion

$$\alpha_i = \sum_{m=0}^{\infty} \frac{\Delta x^m}{m!} \left(\frac{\partial^m \alpha}{\partial x^m} \right)_{i-1} \tag{2.23}$$

one can rearrange for the spatial derivative of Eq. (2.21)

$$\frac{\partial \alpha}{\partial x} = \frac{\alpha_i - \alpha_{i-1}}{\Delta x} - \frac{\partial^2 \alpha}{\partial x^2} \frac{\Delta x}{2} - \frac{\partial^3 \alpha}{\partial x^3} \frac{\Delta x^2}{6} - \cdots \tag{2.24}$$

Now approximating the derivative with a first-order accuracy, by accepting truncation errors of $\mathcal{O}(\Delta x)$, leaves the so-called backward difference

$$\alpha_x \approx \frac{\alpha_i - \alpha_{i-1}}{\Delta x} \ . \tag{2.25}$$

Using the similar difference representation for the time derivative but forward in time, the approximate form of Eq. (2.21) accurate to first-order in space and time is

$$\alpha_t + u\alpha_x \approx \frac{\alpha_i^{n+1} - \alpha_i^n}{\Delta t} + u \frac{\alpha_i^n - \alpha_{i-1}^n}{\Delta x} \ . \tag{2.26}$$

One should note that this corresponds to first-order accurate finite volume discretization of Eq. (2.21) since the situation is approached in 1D with constant convection velocity and grid spacing. Convection of a step profile and a sin-squared wave are given in Fig. 2.2(a). The numerical diffusion inherent in first-order methods is devastating, and completely smears the profiles to unrecognisably smooth humps.

A hasty way of salvation might immediately lurk into one's mind, namely a mere raise in order of the truncation error of the method. For spatial discretization, this can be done using so-called central difference (cf. Fletcher (1991); Siikonen (2008))

$$\alpha_t + u\alpha_x \approx \frac{\alpha_i^{n+1} - \alpha_i^n}{\Delta t} + u \frac{\alpha_{i+1}^n - \alpha_{i-1}^n}{2\Delta x} \ , \tag{2.27}$$

which is second-order accurate in space. This algorithm is however unstable,

and the solution diverges despite the choice of the increments Δt or Δx; it is said to be unconditionally unstable (Siikonen, 2008; Tannehill et al., 1997). A small change in one term results in a complete and utter *u-turn* in the behaviour of the algorithm. It can be stabilized using more implicit information on the right hand side, or

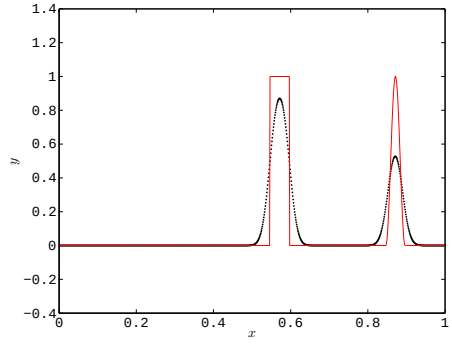
$$\alpha_t + u\alpha_x \approx \frac{\alpha_i^{n+1} - \alpha_i^n}{\Delta t} + u\frac{\alpha_{i+1}^{n+1} - \alpha_{i-1}^{n+1}}{2\Delta x} \ . \tag{2.28}$$

The resulting algorithm is unconditionally stable. However, the results are still completely smeared due to its retained temporal accuracy (Fig. 2.2(b)). Again, this can be raised by one order using so-called Crank-Nicolson method, where averages between the two time levels are used in the spatial derivatives. The computed results are in Fig. 2.2(e).

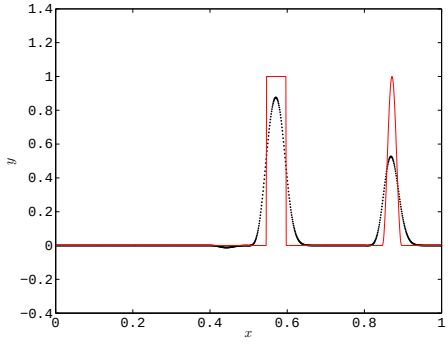
For further comparison, higher-order explicit time integrations can be obtained using the Runge-Kutta method. Results using a fourth-order method are shown in Fig. 2.2(f). Stable higher spatial order methods can be obtained using wider stencils in space and with upwind biasing for unsymmetrical stencils. The algorithm used to compute the results will be detailed later.

Results from applying the above schemes are shown in Fig. 2.2. The least accurate methods result in very diffusive algorithms. This is clear from Figs. 2.2(a) and 2.2(b), where both profiles are smeared beyond recognition. Higher-order approximations provide better approximation to the sin-squared wave. Solutions with the implicit integrations are mildly lagging at the peak (Figs. 2.2(c)-2.2(e)) and somewhat dispersive at either the trailing or leading edge, depending on the discretizations. The opposing edge is again smoothed. These phenomena escalate for the square wave, and the solution is highly oscillatory in the level peak. For the highest-order method given in Fig. 2.2(f), the leading edges come with mild oscillation, but is much lessened. The trailing edges are still scantly smoothed with this algorithm, though the level peak is better resolved, yet still not acceptable. Merely the resolution of the sin-squared peak is tolerable.

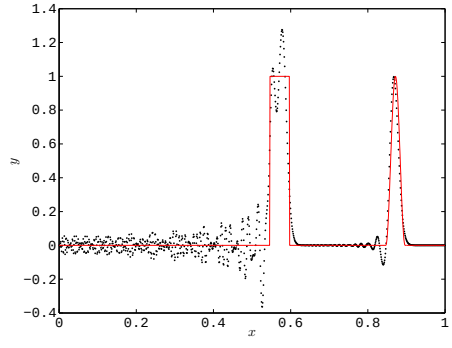
Measures to decrease the truncation error in time serve as somewhat stabilizing as the overshoots are diminished. The direction for dispersion and diffusion changes depending on the combinations between temporal and



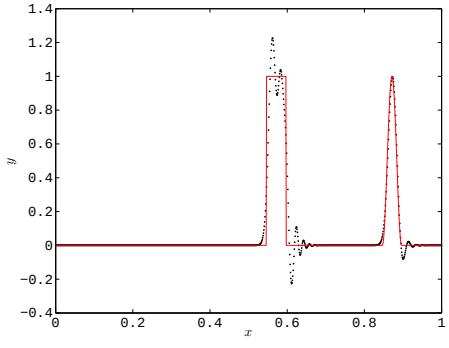
(a) First-order method in space with explicit Euler time integration.



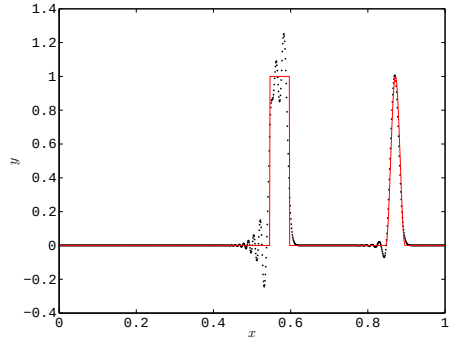
(b) Second-order symmetric method in space with implicit Euler time integration.



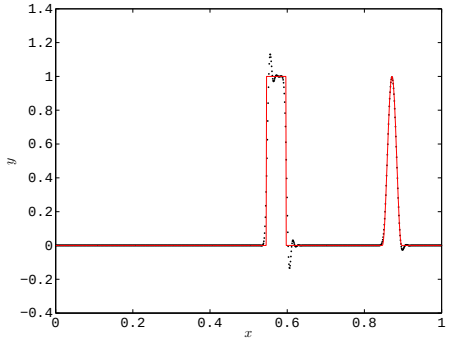
(c) Second-order symmetric method in space with implicit Crank-Nicolson time integration.



(d) Second-order upwind biased method in space with implicit Crank-Nicolson time integration.



(e) Third-order method in space with implicit Crank-Nicolson time integration.



(f) Third-order method in space with explicit four-step Runge-Kutta time integration.

Figure 2.2: Example computations of the convection of step profile and a sin-squared wave. The Courant number $\mathsf{C} \equiv u\Delta t/\Delta x = 0.5$ in all computations. The exact solution is indicated by the red line, and the ordinate $y \equiv \alpha$.

spatial discretizations. Increasing the spatial discretization to a third-order enhances the resolution minutely, but introduces dispersion downstream that evidently spoils the solution.

One should note that the above mix of conceptual examples are qualitative at most. Clearly the situation changes drastically when the spatial derivatives are reconstructed purely from upwind or from upwind and downwind values[3]. This can be seen especially in Figs. 2.2(c)-2.2(f). Also, the temporal discretization has a clear impact on the stability of the method, and the directions for diffusion and dispersion depend also on the time integration.

The problems associated with the approximate solution of the convection equation were illuminated. The pure convection problem is not an easy one to solve numerically. Trivially, the linear problem can be solved as it readily has an exact solution, and provides one also numerically (in 1D) with the known characteristic (convection) velocity. This situation is, however, complicated in multidimensional cases where the computational cells might be skewed and the direction of the convection velocity differs from the alignment of the grid lines. Also, when solving the system of Navier-Stokes equations, Eq. (2.21) is not necessarily linear any longer. Numerical schemes devised in simple circumstances reveal their weak points clearer and provide information upon which each of the problems can be solved. The problem is approached more quantitatively in the following chapters.

Numerical diffusion and dispersion

The features of a discrete scheme can be studied via a so-called modified equation (LeVeque, 1992; Siikonen, 2008; Tannehill et al., 1997). It may be thought of as the partial differential equation that is *actually* solved when a particular difference scheme is applied for the numerical solution of the basic equation. For example, the modified equation for Eq. (2.26) is

$$\frac{u\Delta x}{2}(1-\mathsf{C})\alpha_{xx} - \frac{(u\Delta x)^2}{6}(2\mathsf{C}^2-3\mathsf{C}+1)\alpha_{xxx} + \cdots \tag{2.29}$$

[3]Since the problem was purely *convective*, the directions are unambiguous. That is, if one goes in a direction opposite to that of the convection, it is denoted as going 'upwind' or 'upstream'.

according to Tannehill et al. (1997); $\mathsf{C} = u\Delta t/\Delta x$ is the Courant number. The first term $\frac{u\Delta x}{2}(1-\mathsf{C})\alpha_{xx}$, or more generally second-order (even) derivatives on the right hand side of the modified equation, represents the numerical or artificial viscosity inherent and dominant in all first-order methods. Its highly diffusive influence is especially evident in Figs. 2.2(a) – 2.2(b). However, the first-order methods come with inherent monotonicity, that is, no new extrema are created in the approximate solution. This comes with the expense of solution accuracy, but produces physically meaningful solutions.

The modified equation for the Crank-Nicolson time integration applied to the central difference scheme (Fig. 2.2(e)) is (Tannehill et al., 1997)

$$-\left(\frac{u(\Delta t)^2}{12}+\frac{u(\Delta x)^2}{6}\right)\alpha_{xxx}+\cdots \tag{2.30}$$

that is, it contains no numerical diffusion. Instead, the presence of the odd derivative α_{xxx} in the modified equation reveals itself in dispersive behaviour, visible as oscillations in the numerical solution. Clearly, the result is unacceptable despite its higher formal accuracy. This is inherent to all higher-order methods. In addition, S.K. Godunov has shown in the 1950's that no second or higher-order scheme is monotone (Siikonen, 2008; Tannehill et al., 1997).

The effects of numerical diffusion and dispersion are present in most numerical schemes. It is their interaction in the modified equation that determines the result of using a specific scheme. Clearly, an adequate discontinuity-capturing method needs to be monotone but at the same time the approximation should be higher than first-order accurate in order to dismiss the need of infinitely fine grids to eliminate the inherent diffusion and assert a higher-order truncation error.

Non-uniqueness of the solution

A mathematical aspect of the weak form of the equations is that they sometimes allow for more than one solution to a conservation law with the same initial data. That is, additional information is required if the result is to be called a physically relevant solution; clearly, there can be only one such solution. For most flow cases, smooth solutions can be obtained using the

Navier-Stokes equations because they contain effects of physical diffusion that prevents the formation of discontinuities due to the non-linearity, for example, but the mere convection equation of the void fraction contains only an unsteady and a convective term; it also introduces the discontinuous properties to the former (the contact discontinuity). For numerical methods, only monotone schemes converge to a physically relevant solution, and, as was seen previously, the higher-order ones may choose a non-physical solution. One way to assure the correct solution is to impose the so-called entropy condition[4]. Another way is adding the effects of diffusion to the model. See, *e.g.*, LeVeque (1992), Siikonen (2008), Jasak (1996) and Yee (1989). Possible issues are a result of the omission of some physical aspects of the phenomenon in the model chosen to represent it[5]. Here the case is such that the material interface between the two fluids is seen as a boundary whose thickness approaches zero; in reality it might be seen as a 'transition zone' with very steep gradients. It actually involves such microscopic interfacial physics that depend on intermolecular forces or other details at the interface, such as the presence of surfactant molecules, and these come with time and length scales that are not achievable without significant computer resources (Lafaurie et al., 1994); it is also difficult to infer their significance in the problem at hand. The modelling approach was already discussed. In practical flow calculations with the Navier-Stokes equations, the viscous effects are always present. The non-physicality in this respect may be an issue via the void fraction equation.

2.2.2 Requirements for the sharp resolution of contact discontinuities

The ideal case on VOF strategies would be to reproduce sharp approximations to the discontinuous solutions automatically, that is, without the need of explicit tracking or the use of separate jump conditions. It is evident from aforementioned information that standard discretization methods either reduce the step profile into a smooth hump and smear it across multiple cells, or introduce oscillations near the discontinuities and steep gradients. Either of these spoils the solution. Based on the observations, one can list

[4]From the second law of thermodynamics, the entropy should be a nondecreasing quantity

[5]Cf. the Euler equations

requirements from the methods to solve Eq. (2.20) approximately:

1. Minimizing numerical diffusion
2. No spurious oscillations in the solution
3. No violations of the physical bounds of the dependent variable
4. At least second-order accuracy in smooth regions

These can overlap one another, of course. Judging from Sec. 2.2.1 only the highly diffusive first-order methods are monotone, *i.e.*, they do not produce new extrema to the solutions. In fact, they could be used locally near the discontinuities to uphold the monotonicity. The degree of this usage dictates the smearing of the solution, and that can be countered by introducing negative artificial viscosity, or compression in the discretization; one must exercise caution since excessive negative diffusion is non-physical. Such discretization schemes are non-linear, that is, they depend on the local solution. The requirement of at least second-order accuracy, whilst preserving non-oscillatory solutions, are called high resolution methods. It is thus desirable to base the discretization of Eq. (2.20) on some kind of a combination with the schemes preserving the monotonicity and boundedness, as well as schemes with sufficient accuracy in the smooth regions. These types of discretization schemes can be devised using different strategies, such as total variation diminishing (TVD) criteria or convective boundedness criteria (CBC). These are considered in Sec. 3.3.2.

The problems encountered above were mainly related to situations in one spatial dimension. Whilst also present in multi-dimensional computations, there are additive issues that need consideration when extending the calculations in 2D, for example. If a discrete mesh is laid atop a water wave as depicted in Fig. 2.1, water can enter, or conversely air can leave (and vice versa), from a computational cell over the four faces. While Eq. (2.20) is in the conservative form, the interface itself does not necessarily align with the cell faces, or cells may be partially filled. In a computer code the fluxes are usually calculated sequentially, that is, one performs a sweep in the i index direction first, after which the procedure is repeated for the j and k index directions for three-dimensional situations. This is possible using structured grid systems, where each index direction can be viewed as

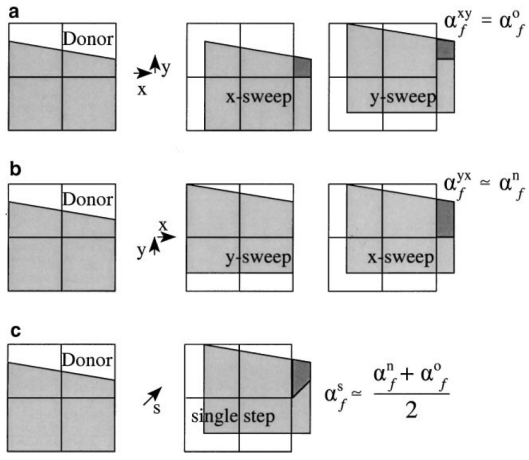


Figure 2.3: Prediction of the cell face value in multiple dimensions. The darker shaded area shows the amount of the dark fluid that has crossed the right face of the donor cell during one time step. Figure taken from Ubbink and Issa (1999).

a one-dimensional case. The cell face values are calculated sequentially, in isolation of each other, and the flux balance is predicted from these. An issue can be illustrated with Fig. 2.3. While the final amount of the fluid convected over the right face of the donor cell is the same in all cases, the intermediate face value is different, depending upon the order of the sweeps in the x and y-directions. Most of the special discretization schemes that are formulated to encompass one or more of the above features were devised for one-dimensional time-dependent problems. At present it is difficult to gather whether the severity is of a substantial nature. It should be noted that the aforementioned strategies have been successfully used in many practical problems.

2.3 The free surface boundary conditions

Flows with free surfaces involve a boundary between two fluids. The time evolution of the boundary follows from the behaviour of the two fluids on both sides of it. For the interpretation of the boundary to be an actual free surface such as an air-water interface, certain conditions must be satisfied. It is important to note that this is true regardless of the methodology used

in the solution of it. Here, the surface tension is neglected, both fields are considered as incompressible and completely segregated. This eases the casting of the conditions, but introduces other headaches. The applicable boundary conditions are (Ferziger and Perić, 1999; Matusiak, 2005):

1. **The kinematic boundary condition**, that is

$$\vec{V} \cdot \nabla\alpha = \vec{V} \cdot \vec{n}_{fs} = 0 \tag{2.31}$$

   on the free surface (fs), since the gradient of the void fraction is the normal ($\vec{n}_{fs}$) for that surface. The velocity ($\vec{V} = \vec{V}_{fluid} - \vec{V}_{fs}$) component normal fro the free surface is the convection velocity of that surface, and the mass flow through the surface is zero. This means that the interface is a streamline in steady-state situations.

2. **The dynamic boundary condition**, which denotes that the forces acting on the fluid at the free surface are in equilibrium. The normal component of the forces acting on one side of the free surface is the same in magnitude but opposite in sign to that on the other side of the surface.

The enforcement of the first condition in a numerical solution is not a straightforward task. It requires that the mass is strictly conserved, or the velocity field is strictly solenoidal, within the iteration; the α equation can be seen as a special case of the mass balance to segregated two-phase flows with no mass transfer. This can be illustrated with Fig. 2.4, where the evolution of the interface is depicted between time instants t and $t+\Delta t$. The evolution of the contour of $\alpha = 0.5$, which is usually considered as a representation of the free surface, is convected by the velocity $\vec{c}$. This velocity follows from the Rankine-Hugoniot argument and it is the convection velocity $\bar{u}$ (Lafaurie et al., 1994). The 'wave front' merely moves with the same velocity, and there is no transfer of mass between the phases through this interface. The determination of this exact contour (the wave front) is not possible with the VOF approach. All mixture properties are averaged in the computational cell, and there is never really a surface present in mixture model approaches. Instead, the interface should somehow be determined by the scalar values of the void fraction in the cells. (It might seem that the single viable option is to interpret the interface to be present at the cell faces *only*). The continuity

equation for the k:th field can be seen as

$$\frac{D\rho_k}{Dt} = -\frac{\rho_k}{\alpha_k}\left(\frac{\partial \alpha_k}{\partial t} + \frac{\partial \alpha_k u_j}{\partial x_j}\right) \tag{2.32}$$

with summation implied only over j. This indicates that an inaccurate

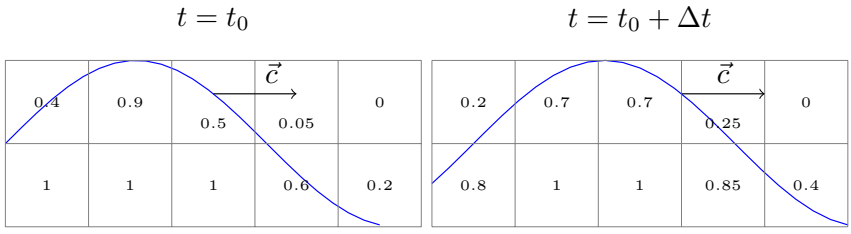


Figure 2.4: Time evolution of an interface.

solution for the equation of the void fraction results in a source of the mass in the iterative solution. This violates the physics of the situation and could spoil the results, possibly leading to a divergent method. The numerical solution of the latter term is discussed in Sec. 3.3. For a numerical solution to the two-phase flow methodology taken here, it could only be said that the interface should be resolved as sharply as possible, preferably to a thickness of only a few cells. The convection equation for the void fraction can thus be seen as the kinematic boundary condition (cf. Hong and Walker (2000), Ferziger and Perić (1999), Vaz et al. (2009) and Montazeri (2010), and Sec. 2.2). This facilitates the study of the discretization practises commonly used in VOF solution strategies, forming a part of the computations done in Chap. 4. How one reaches this goal with a computer code is then again not a trivial task, as was demonstrated earlier.

The second condition is accompanied with its owns issues. Applying the momentum equations for the set-up of Fig. 2.5 leaves us with

$$\nabla p = \rho \vec{g} \ , \tag{2.33}$$

that is, the pressure gradient term remains due to the gravity source. This gives rise to the term *hydrostatic* pressure, which will be discussed later. The expression can be integrated in a one-dimensional example shown in Fig. 2.5

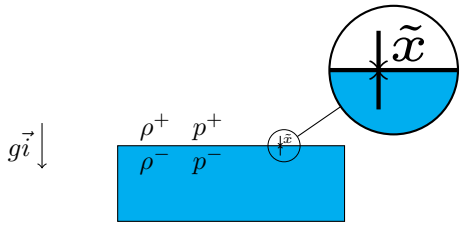


Figure 2.5: An interface of two fluids at rest.

across the interface

$$\begin{aligned} \int_{p^-}^{p^+} dp &= \int_{x^-}^{x^+} \rho g \, dx \\ \rightarrow (p^+ - p^-) &= (\rho^+ - \rho^-) g \tilde{x} \ , \end{aligned} \tag{2.34}$$

where $\tilde{x}$ is an infinitesimal thickness of the interface, and the superscripts $+/-$ denote the values on the upper and lower sides of it, respectively. A rather philosophical question emerges, namely the differential form of the equations ceases to have a unique solution in the classical sense across the interface. The integral form however remains valid. Then again, if the interface has a finite thickness the dynamic boundary condition is not fulfilled, *i.e.*, it is satisfied only at the limit of $\tilde{x} \rightarrow 0$. One should remember that within this approach the interface can never truly be a distinct surface, but merely a distribution of α (or the mixture properties) that follows from the local flow solution. The void fraction is a multi-valued function across the interface, the free surface itself is usually (artificially) defined as the iso-surface of $\alpha = 0.5$. Clearly, this problem is not very well posed. In order to be more realistic, this pressure jump at the interface should be balanced by the surface tension on the interface through the Young-Laplace relation

$$\Delta p = \kappa \sigma \ , \tag{2.35}$$

where Δp is the pressure difference across the interface, κ is the local curvature of the interface and σ is the constant coefficient of surface tension (Montazeri, 2010). The appropriate discrete formulation of this could satisfy the dynamic boundary condition, and it appears as another source term in the momentum equations (2.2). In fact, it is possible to formulate this relation such that it is evaluated only at the iso-surface $\alpha = 0.5$ and so as to

enforce the dynamic equilibrium (Nandi and Date, 2009). In this work, the surface tension is considered negligible as is the common practice in naval hydrodynamics. It hardly ever influences the momentum balance for flows that would be of general interest to a naval architect. Now the dynamic boundary condition needs special attention in this respect. Problems can be circumvented again by declaring the existence of the interface only at the cell faces, but in an approximate solution this may not be the case. Also the grid alignment should be somehow *adjusted* (cf. the interface tracking approach!). This gives rise to so-called surface capturing methods that use reconstruction of the free surface shape. Queutey and Visonneau (2007) proposed a way to address the pressure jump without addition of the surface tension, instead formulating an improved expression for the discretized cell face pressures and densities; it was argued that this practice is the only choice. Hänninen and Mikkola (2007) implemented those formulations in the code YAFFA, and they are given later. Montazeri (2010) and Montazeri et al. (2012, 2014) continue on the formalism of Hong and Walker (2000), that is incorporating the dynamic boundary condition directly into the momentum equation by the use of the piezometric pressure $p + \rho g z$ in the equations, but the model requires phasic pressures. Orych (2013), Wackers and Koren (2005) and Wackers (2007) define the input states of the hydrostatic contribution as averages in their approximative Riemann solvers, cancelling the effects of the discontinuous density in a quiescent fluid. Such an approach stems from the source term balancing method, introduced by, *e.g.* Leveque (1998) and Hubbard and Carcia-Navarro (2000). Qian et al. (2006) split the pressure gradient normal to the initial still water level to kinematic and hydrostatic terms and exchange the hydrostatic pressure gradient directly to the local density values in cell centres in the normal direction. They discuss that the treatment then exactly balances the gravity source term; this seems to be the same approach as above.

In an incompressible situation an effective pressure must be used

$$p = p_{dif} + p_{\infty} + \rho \vec{g} \cdot \vec{x} \ , \tag{2.36}$$

where p is the pressure, p_{dif} is the pressure difference, p_{∞} a reference pressure and the last term denotes the hydrostatic part that stems from Eq. (2.33). $\vec{x}$ is the height from a reference level. This definition of the pressure difference is necessary since the pressure level is undetermined *a priori*. The

pressure affects the momentum balance only through its gradients. It is usually convenient to determine the initial free surface level with $\vec{x} = (x, y, 0)$. That is, the hydrostatic component is zero on that level. To uphold this physicality, one should set $p_\infty = p_{atm}$ at the standard sea level to remember the *aero*static contribution. This definition is also convenient in the solution procedure since the constant ambient pressure can be eliminated. That is, one calculates instead of the pressure the term called pressure difference and obtains solutions where the pressures are those relative to this constant value. This has also accuracy considerations, since the p_{dif}, or a pressure disturbance caused by an obstacle altering the flow field, is in most cases smaller than the pressure by several orders of magnitude. In many situations, the hydrostatic contribution can be neglected since it provides a constant source term (with constant ρ and $\vec{g}$) for the momentum equations (Matusiak, 2005; Newman, 1977; Tannehill et al., 1997). That is, the same forcing takes place throughout the solution, or the effects of it are the same for all computational cells and thus can be omitted. Now the gradient of this pressure is actually

$$\nabla p = \nabla p_{dif} + \rho\vec{g} + \vec{g} \cdot \vec{x}\ \nabla\rho\ , \tag{2.37}$$

that is, gradient of density appears on the interface. Integrating Eq. (2.37) for a control volume $\mathcal{V}$ gives

$$\int\limits_{\partial\mathcal{V}} p\ d\vec{S} = \int\limits_{\partial\mathcal{V}} p_{dif} + \rho\vec{g} \cdot \vec{x}\ d\vec{S} + \int\limits_{\mathcal{V}} \rho\vec{g}\ d\mathcal{V}\ . \tag{2.38}$$

Instead of the mere pressure difference, strictly speaking, the contribution from the density gradient should be seen in the faces of the control volume in regions where it contains the mixture. The use of pressure differences in the solution procedure would require reformulation based on Eq. (2.37). This seems to be a rather new approach. It has been accounted for by Rusche (2002), and applied by him, Kissling et al. (2010), Berberović (2010) and Paulsen (2013). Some possible implications will be discussed later. This problem can again be dodged if the interface appears only on the cell faces; then there is no gradient of density and the additional face term vanishes.

A short summary is in order. The kinematic boundary condition in the VOF-methodology is related to the solution of the α equation as well as to the divergence-free velocity field. The dynamic boundary condition is in this

case related to the pressure term in the momentum equations. These share a strong coupling, as will be seen later. The former BC is studied through a few solution strategies of the α equation, and the pressure correction stage which is driven by the mass (or volumetric) balance. For the latter BC, the approach of Queutey and Visonneau (2007) was implemented in the reference code, whereas FINFLO uses simple averages for the cell face pressures. The fulfilment of the free surface boundary conditions will be studied within the computations made in this thesis, and notion is made in this respect in all relevant sections concerning the above remarks.

Chapter 3

Numerical method

3.1 Flow solvers

In this section, the codes used to solve the two-phase flow case are presented. Both solvers use a control volume method to discretize the governing flow equations but differ on the implementation of the two-phase models, for instance. Such differences are fundamental, and lead to a myriad of solution methods employed in one code that is completely outside of the realm of the other. However, a physical phenomenon that is of the academic-test type is the goal of the quest here, and it is resolvable using either codes.

This chapter presents the unstructured and structured control volume methods used to discretize the governing flow equations, and the resulting sets of algebraic equations. This is done in subsequent sections. Special features of both codes and their modelling method for two-phase flows are described. The code Yaffa is used as a reference, and the description of it is left more superficial where this comes with no hindering consequences.

3.1.1 Yaffa

The program Yaffa is a two-dimensional flow solver developed at the Helsinki University of Technology (currently Aalto University) and described *e.g.* in

a PhD Thesis by Mikkola (2009). It is a Navier-Stokes solver with the SIMPLE-type pressure correction method. YAFFA excludes any turbulence models. The solver readily includes the free surface tracking method (cf. Sec. 1.3) for simulations of free surface flows. Mikkola (2009) used this approach in simulating flows over a NACA-0012 profile and a Gaussian ground elevation. The code had also been developed to include a free surface capturing capability, namely through the VOF-approach (Hänninen and Mikkola, 2007). That development forms a part of the basis for this thesis.

3.1.2 Finflo

FINFLO (Finflo, 2013) is a general-purpose CFD code developed at Helsinki University of Technology and currently maintained by Finflo Ltd. FINFLO uses a multi-block structured control volume method for the discretization of the computational domain. The equations can be solved with either the pseudo-compressibility method or a SIMPLE type pressure correction method. Parallelization is done using the MPI-standard. The code has been applied for numerous computations of flows around ships (cf. Li and Matusiak (2001); Li et al. (2001)), and it currently has the capability to solve the free surface problem using a deforming grid technique. This technique was also applied to the present case by Mikkola (1999). Recently, a cavitation model has been implemented (Miettinen et al., 2006; Siikonen, 2009; Siikonen et al., 2012), and applied for computations of cavitating propeller flows (Sipilä, 2012). That development forms a part of the basis for this thesis.

3.2 Discretization of the flow equations

The governing flow equations form a set of non-linear partial differential equations. Few analytical solutions exist (cf. White (2006)), but each are for rather simple cases that are not of considerable engineering use. If one seeks their solution to a problem that may be of interest to a naval architect (cf. Sec. 1.2), the only option is still to solve them with an acceptable level of approximation. It is common practice to achieve this numerically using a computer code that solves Eqs. (2.1) and (2.2), amongst possible auxiliary relations, *e.g.* for turbulence closure. However, for an exploitation of a

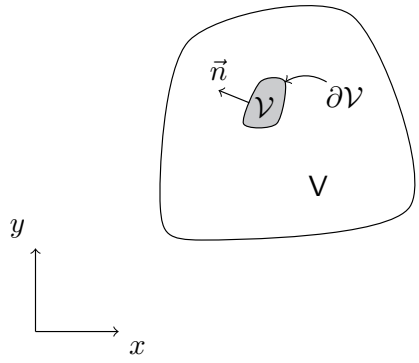


Figure 3.1: A general control volume $\mathcal{V}$ inside the computational domain V. $\partial\mathcal{V}$ is the bounding surface of $\mathcal{V}$ and $\vec{n}$ its normal vector.

computer the governing equations must be cast to a sequence of algebraic procedures. This is achieved through the process of discretization.

It is here again noted that both the standard form and the component form of the vector notation are used in this work, or $\vec{u}$ and u_i, respectively. However, mostly in description of the numerical schemes in the structured grid systems, the index i denotes the i:th computational cell. In description of a differential equation, or a numerical scheme in unstructured grid systems, the index i denotes the i:th component. These should be unambiguous, but a notion is made when tensor procedures are used.

In this work, the governing equations are discretized with the finite volume method (cf. Ferziger and Perić (1999); Siikonen (2008); Tannehill et al. (1997)). The computational domain V is split into i subdomains $\mathcal{V}$ *s.t.* $\sum_i \mathcal{V}_i = \mathsf{V}$, see Fig. 3.1. Variables are defined at the centres of these subdomains (computational cells). Now the discretization is based on the integral form of the governing flow equations, which is valid for both laminar and turbulent flow. These are

$$\frac{\partial}{\partial t}\int_{\mathcal{V}} \rho \, d\mathcal{V} + \int_{\partial\mathcal{V}} \rho\vec{V}\cdot d\vec{S} = 0 \tag{3.1}$$

for the conservation of mass, where ρ is the density of the mixture, $\vec{V}$ its velocity field and $d\vec{S} = dS\vec{n}$ the oriented surface area element of the control volume $\mathcal{V}$. The mixture properties are conditioned with the void fraction;

this procedure was readily discussed in Sec. 2.2. The balance of momentum reads

$$\frac{\partial}{\partial t}\int\limits_{\mathcal{V}} \rho\vec{V}\ d\mathcal{V} + \int\limits_{\partial\mathcal{V}} \rho\vec{V}\vec{V}\cdot d\vec{S} = \sum \vec{f}\ . \tag{3.2}$$

This equation states that the time rate of change of momentum $\rho\vec{V}$ in $\mathcal{V}$ results from its convective fluxes through $\partial\mathcal{V}$ and from the sum of the vector $\vec{f}$, which includes the forces applied unto $\mathcal{V}$ or its bounding surfaces. Here, we are only interested in surface forces due to pressure and stress, and volume forces due to gravity, *i.e.*

$$\vec{f} = \int\limits_{\partial\mathcal{V}} -p\vec{n} + \tau_{ij}\cdot\vec{n}\ dS + \int\limits_{\mathcal{V}} \rho\vec{g}\ d\mathcal{V}\ , \tag{3.3}$$

where p is the pressure, τ_{ij} are the components of the viscous stress tensor and $\vec{g}$ the acceleration due to gravity. If the conservation of mass and momentum are satisfied for each individual cell, they are satisfied throughout the computational domain. This means that they are automatically conserved in a numerical solution as well (Tannehill et al., 1997).

The two codes used here differ somewhat in the discretization and solution procedures used. This is taken into account in the organization and structure of the following subsections, where the discretization of each term of Eq. (3.2) is described. For the sake of brevity, the governing equations can be cast to a general formulation, including the differential and integral forms as

$$\frac{\partial\vec{U}}{\partial t} + \frac{\partial(\vec{F}-\vec{F}_v)}{\partial x} + \frac{\partial(\vec{G}-\vec{G}_v)}{\partial y} + \frac{\partial(\vec{H}-\vec{H}_v)}{\partial z} = \vec{Q} \quad \text{and} \tag{3.4}$$

$$\frac{\partial}{\partial t}\int\limits_{\mathcal{V}} \vec{U}\ d\mathcal{V} + \int\limits_{\partial\mathcal{V}} \left(\vec{F}-\vec{F}_v+\vec{G}-\vec{G}_v+\vec{H}-\vec{H}_v\right)\cdot d\vec{S} = \int\limits_{\mathcal{V}} \vec{Q}\ d\mathcal{V} \tag{3.5}$$

where the vector $\vec{U}$ consists of the variables to be solved, $\vec{F}$, $\vec{G}$ and $\vec{H}$ are the inviscid fluxes in x, y and z-directions, respectively, and $\vec{F}_v$, $\vec{G}_v$ and $\vec{H}_v$ the corresponding viscous fluxes. Vector $\vec{Q}$ contains the possible source terms. These are given explicitly below.

Eq. (3.5) can be approximated for a computational cell i as

$$\mathcal{V}_i \frac{d\vec{U}_i}{dt} = \sum_{faces} -\hat{F}S + \mathcal{V}_i Q_i \ , \tag{3.6}$$

where the fluxes on the cell faces are

$$\hat{F} = n_x(F - F_v) + n_y(G - G_v) + n_z(H - H_v) \tag{3.7}$$

with the surface normals n_i in each direction i.

Flow equations in Yaffa In this study, the code YAFFA solves the incompressible Euler equations in two-dimensions. The force vector on the right hand side of Eq. (3.2) reduces to

$$\vec{f} = \int_{\partial\mathcal{V}} -p/\rho \ \vec{n} \ dS + \int_{\mathcal{V}} \vec{g} \ d\mathcal{V} \ , \tag{3.8}$$

that is, the viscous stress tensor τ_{ij} disappears, and only the pressure and contribution from the acceleration due to gravity retain. To utilize the system (3.4), the vector of unknown variables is

$$\vec{U} = (u, v, \alpha)^{\mathsf{T}} \ , \tag{3.9}$$

where α is the void fraction. The inviscid fluxes in the x and y-direction are

$$\vec{F} = \begin{pmatrix} uu + p/\rho \\ uv \\ \alpha u \end{pmatrix}, \quad \vec{G} = \begin{pmatrix} vu \\ vv + p/\rho \\ \alpha v \end{pmatrix} \ , \tag{3.10}$$

and the source terms

$$\vec{Q} = (g_x, g_y, 0)^{\mathsf{T}} \ , \tag{3.11}$$

where g_i are the components of the vector of gravitational acceleration. As the incompressible limit was shown to hold in this variable density flow in Sec. 2.2, the system can now be divided by the mixture density; a fundamental aspect of the problem at hand. This will be detailed later. Unless otherwise stated, the following sections considering the single-phase methodology of the code YAFFA are based on the PhD thesis of Mikkola (2009). In this work, the documentation is extended based on the work of Hänninen

and Mikkola (2007) to the implemented VOF-algorithm.

Flow equations in Finflo The code FINFLO solves the Reynolds-averaged Navier-Stokes equations for three-dimensional flow cases. In two-equation turbulence models, the scalar transport equations for the kinetic energy k and dissipation ε (or the specific dissipation, ω) of turbulence are included. In two-phase simulations, the scalar transport equation for the void fraction α is solved interchangeably with the gas mass fraction, x_g. The general vector of unknown variables is

$$\vec{U} = (\rho, \rho u, \rho v, \rho w, E, \rho k, \rho\varepsilon, \rho\phi)^{\mathsf{T}} \ . \tag{3.12}$$

The inviscid fluxes are

$$\vec{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p + \frac{2}{3}\rho k \\ \rho v u \\ \rho w u \\ u(E + p + \frac{2}{3}\rho k) \\ \rho u k \\ \rho u \omega \\ \rho u \phi \end{pmatrix}, \ \vec{G} = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p + \frac{2}{3}\rho k \\ \rho w v \\ v(E + p + \frac{2}{3}\rho k) \\ \rho v k \\ \rho v \omega \\ \rho v \phi \end{pmatrix}, \ \vec{H} = \begin{pmatrix} \rho w \\ \rho u w \\ \rho v w \\ \rho w^2 + p + \frac{2}{3}\rho k \\ w(E + p + \frac{2}{3}\rho k) \\ \rho w k \\ \rho w \omega \\ \rho w \phi \end{pmatrix} . \tag{3.13}$$

ρ is the mixture density and $\vec{V} = u\vec{i}+v\vec{j}+w\vec{k}$ is the three-dimensional velocity vector, and ϕ a possible additional scalar variable. The total internal energy is

$$E = \rho e + \frac{\rho\vec{V}\cdot\vec{V}}{2} + \rho k \ , \tag{3.14}$$

where e is the specific internal energy. This is however not utilized in the current cavitation model, and the energy residuals are set to zero. The viscous fluxes are

$$\vec{F}_v = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x \\ \mu_k\,(\partial k/\partial x) \\ \mu_\omega\,(\partial\omega/\partial x) \\ 0 \end{pmatrix}, \ \vec{G}_v = \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} - q_y \\ \mu_k\,(\partial k/\partial y) \\ \mu_\omega\,(\partial\omega/\partial y) \\ 0 \end{pmatrix},$$

$$\vec{H}_v = \begin{pmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} - q_z \\ \mu_k \left(\partial k/\partial z\right) \\ \mu_\omega \left(\partial \omega/\partial z\right) \\ 0 \end{pmatrix}, \tag{3.15}$$

where the viscous stress tensor is familiarly

$$\tau_{ij} = \mu \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \left(\rho \overline{u_i' u_j'} - \frac{2}{3} \rho k \delta_{ij} \right) , \tag{3.16}$$

and δ_{ij} is the Kronecker delta. The turbulent diffusion coefficients are

$$\mu_k = \mu + \frac{\mu_t}{\sigma_k} , \quad \mu_\omega = \mu + \frac{\mu_t}{\sigma_\omega} , \tag{3.17}$$

where σ_i are the turbulent Schmidt numbers and μ_t the turbulent viscosity of the fluid. q_i are the heat fluxes. The source term has non-zero components for the momentum equations due to the gravity, and for the turbulence closures. Unless otherwise stated, the following sections considering the code FINFLO are based mainly in Ref. Finflo (2013).

3.2.1 Solution algorithm

Yaffa The basic solution algorithm is based on the pressure correction method of the SIMPLE type. The equations are solved in a segregated manner, and the procedure is started by computing the velocity components. The procedure is depicted in Fig. 3.2. The solution is based on finding a flux balance for each control volume. The explicit residual is the balance of the non-conservative momentum fluxes in a control volume. The fluxes are of the convective type, and described in Sec. 3.2.2. The linearizations of the momentum equations is based on the first-order upwind method, and form the basis for the implicit stage. The explicit residual is the driving force of the implicit stage. After solving for the change in the velocity components, the error in the balances of *volumetric* fluxes is calculated based on the intermediate convection velocity, and that forms driving force for the pressure correction equation. These steps form the inner loop, and are repeated $\mathcal{O}(20)$ times in order to obtain a sufficiently converged solution. After this, the void fraction equation is solved in a similar manner as is

described in Sec. 3.3 with the exception that in YAFFA, a fully implicit three-level discretization of the time derivative is used.

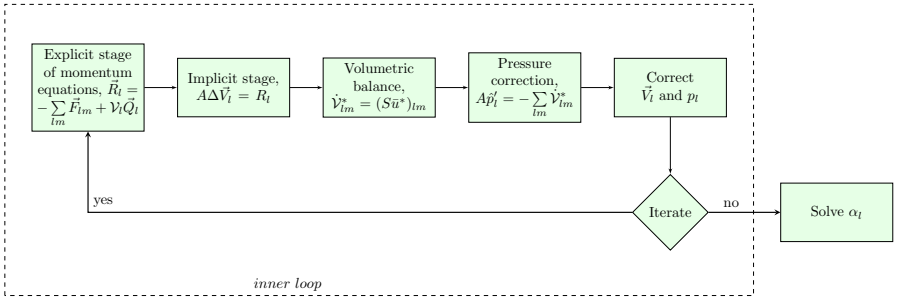


Figure 3.2: Solution procedure of one iteration in YAFFA. The solution of the void fraction α_l bears resemblance to that described in Sec. 3.3.

Finflo The solution algorithm is based on the pressure correction method of the SIMPLE type. Recently, some novel features have been included in the algorithm. These features resemble those used in traditional density-based methods. In density-based methods, the equation system is coupled and solved as such, and in FINFLO this is done by transforming the conservative variables into characteristic ones, with which the system reduces into a tridiagonal set of equations. In the pressure-based solution sequence of FINFLO, the residuals of the unknown variables are calculated only once during an iteration cycle; in traditional pressure correction methods this is done sequentially and in various parts of the procedure, as can be seen above. This corresponds to the transformation matrix used in density-based methods to change the residuals of the conservative variables into those of primitive variables (and those would be then transformed to the characteristic variables).

The momentum residuals are transformed through

$$\rho\Delta\vec{V} = \Delta(\rho\vec{V}) - \vec{V}\Delta\rho \ . \tag{3.18}$$

The same procedure is done for the scalar variables. The energy residuals are more complicated, but luckily of no use here since they are set to zero before the implicit stage. The error in mass balance from the previous iteration cycle $\Delta\rho$ must be eliminated from the residuals, since otherwise the equations must be solved coupled to each other. Now this does not add to the

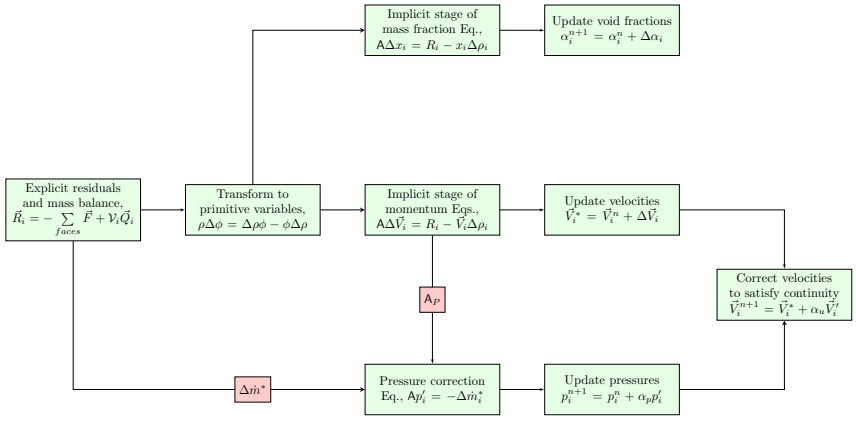


Figure 3.3: Solution procedure of one iteration in FINFLO. The solution of the void fraction α, or gas mass fraction x_g, is described in Sec. 3.2.6, and the pressure correction stage is described in Sec. 3.2.5.

error made in the calculation of momentum balance. Siikonen (1987) used a similar procedure for the solution of the energy equation. In this approach the complexity of the coupled set of equations is avoided by manipulating the explicit residuals. (Siikonen, 2009, 2011)

The solution is based on finding a balance in the fluxes for each control volume, and the procedure depicted in Fig. 3.3. In addition to the momentum equations (Eq. 3.2) for the mixture, an equation for the evolution of the interface is required. In the homogeneous mixture model of FINFLO, this is written with the gas mass fraction x_g

$$\frac{\partial \rho x_g}{\partial t} + \frac{\partial x_g \rho u_i}{\partial x_i} = \Gamma \ , \tag{3.19}$$

where the mass fractions are related to the void fraction through $x_g\rho = \alpha\rho_g$ and Γ is the mass transfer term between the phases; summation is implied over i. Obviously, now $\Gamma = 0$. The use of the mass fractions is convenient in the implicit stage, as the mixture density is used (Siikonen, 2009). The void fractions are used in calculation of the convective fluxes.

The relevant field equations in their semi-discretized form for control

volume i are

$$\mathcal{V}_i \frac{d\rho_i x_{g,i}}{dt} + \sum_j \rho_j \bar{u}_j x_{g,j} S_j = 0 \quad \text{and} \tag{3.20}$$

$$\mathcal{V}_i \frac{d\rho_i \vec{V}_i}{dt} + \sum_j \rho_j \bar{u}_j \vec{V}_j S_j + \sum_j \vec{n}_j p_j S_j = \sum_j \tau_{ij} \cdot \vec{n}_j + \mathcal{V}_i(\rho_i - \rho_\infty)\vec{g} \ , \tag{3.21}$$

where the sum over j denotes summation over all the faces S_j bounding the control volume i, and $\bar{u} = \vec{V} \cdot \vec{n}$ is the convection velocity. These can be written with more concise notation

$$\mathcal{V}_i \frac{\Delta \vec{U}_i}{\Delta t} = - \sum_{faces} \vec{F}^{n+1} S + \mathcal{V}_i \vec{Q}_i \ , \tag{3.22}$$

where $\vec{U}_i$ is the vector of unknown variables (Eq. (3.12)), $\vec{F}$ are the flux vectors on the cell faces, and $\vec{Q}_i$ the source terms. The fluxes are expressed in the new time level $n+1$ and hence need linearizing; otherwise the equations cannot be solved. This is done for all terms as

$$\vec{F}^{n+1} = \vec{F}^n + \frac{\partial \vec{F}}{\partial \vec{U}} \Delta \vec{U} \ . \tag{3.23}$$

All the linearizations are based on the first-order upwind type for standard convection-diffusion equation.

The diffusion fluxes in the momentum equations include derivatives on the cell face (cf. Sec. 3.2.3). However, the use of the full diffusion term in the implicit stage results in a fuller matrix (Miettinen et al., 2006), *i.e.*, the solution of the linear equations is more time consuming. In FINFLO, the linearization of the diffusion flux is based on the expression

$$\vec{F}_{v,i+1/2} = - \left(\frac{\mu}{\Delta x} \right)_{i+1/2} (\vec{U}_{i+1} - \vec{U}_i) \ , \tag{3.24}$$

whence

$$\vec{F}^{n+1}_{v,i+1/2} = \vec{F}^n_{v,i+1/2} + \frac{\mu_i}{\Delta x} \Delta \vec{U}_i - \frac{\mu_{i+1}}{\Delta x} \Delta \vec{U}_{i+1} \ , \tag{3.25}$$

where μ is the diffusion coefficient and Δx is the distance between the centre points of cells i and $i+1$. The convective terms at the cell face for the gas

mass fraction and momentum are

$$\int_{\partial\mathcal{V}} x_g \rho \vec{V} \cdot d\vec{S} \approx \sum_{faces} S_j \rho_j \bar{u} x_g = \sum_{faces} \dot{m} x_g \quad \text{and} \tag{3.26}$$

$$\int_{\partial\mathcal{V}} \rho \vec{V} \vec{V} \cdot d\vec{S} \approx \sum_{faces} S_j \rho_j \bar{u}_j \vec{V} = \sum_{faces} \dot{m} \vec{V} \ , \tag{3.27}$$

that is, the mass flux of the mixture $\dot{m}$ on the cell faces can be identified in the equations. The calculation of the convective terms on the cell faces is detailed in Sec. 3.2.2. The solution of the mass fraction (and α) is resumed in Sec. 3.2.6.

The Jacobians for the convection and diffusion fluxes in the momentum equations, Eqs. (3.13 – 3.15), multiplied by the respective change, are approximated with expressions of the type

$$\left.\frac{\partial F}{\partial U}\Delta U\right|_{i+1/2} = \mathsf{max}(\dot{m}_{i+1/2}, 0)\Delta u_i + \mathsf{max}(-\dot{m}_{i+1/2}, 0)\Delta u_{i+1} \ , \tag{3.28}$$

$$\left.\frac{\partial F_v}{\partial U}\Delta U\right|_{i+1/2} = \left.\frac{S\mu_{eff}}{\Delta x}\right|_{i+1/2} \cdot (\Delta u_i - \Delta u_{i+1}) \ , \tag{3.29}$$

where the max function is utilized to account for the flow direction, $\mu_{eff} = \mu + \mu_t$ and Δx is the distance between the centre points of the cells i and $i+1$. These give the implicit stage in two-dimensional structured grids (Siikonen, 2011)

$$\begin{aligned}
\mathsf{A}_W &= -\mathsf{max}(\dot{m}_{i-1/2}, 0) - \left(\frac{S\mu_{eff}}{\Delta s}\right)_{i-1/2} \ , \\
\mathsf{A}_E &= -\mathsf{max}(-\dot{m}_{i+1/2}, 0) - \left(\frac{S\mu_{eff}}{\Delta s}\right)_{i+1/2} \ , \\
\mathsf{A}_S &= -\mathsf{max}(\dot{m}_{j-1/2}, 0) - \left(\frac{S\mu_{eff}}{\Delta s}\right)_{j-1/2} \ , \\
\mathsf{A}_N &= -\mathsf{max}(-\dot{m}_{j+1/2}, 0) - \left(\frac{S\mu_{eff}}{\Delta s}\right)_{j+1/2} \quad \text{and} \\
\mathsf{A}_P &= \frac{\mathcal{V}_i \rho_i}{\Delta t} - \mathsf{A}_W - \mathsf{A}_E - \mathsf{A}_S - \mathsf{A}_N
\end{aligned} \tag{3.30}$$

for the u-momentum, for instance. Two additional arrays A_T and A_B are needed in three-dimensional calculations. The set of linear equations for

each control volume is

$$\mathsf{A}\Delta u = R \ , \tag{3.31}$$

where R denotes the recalculated residual shown in Eq. (3.18), added together with the contribution of the cell face pressure in the inviscid fluxes as well as with the source term. The cell face pressures are calculated as averages from the nodes surrounding the face, and the contribution from the gravity source is discussed in Sec. 3.2.4. The pressure correction stage is described in Sec. 3.2.5, and the void fraction equation in Sec. 3.2.6. The discretized equations are solved using the DDADI-factorization (diagonally dominant alternating direction implicit).

3.2.2 Convection term

Yaffa The inviscid fluxes are

$$\vec{F}_{lm} = \left(\bar{u}\vec{V} - p\vec{n}/\rho\right)_{lm} S_{lm} \tag{3.32}$$

for the bulk flow variables and

$$F_{lm} = (\bar{u}\alpha)_{lm} \, S_{lm} \tag{3.33}$$

for the void fraction. Above, the index lm denotes the face between the cells l and m, see Fig. 3.4. Familiarly, $\bar{u}_{lm}$ is the convection velocity and S_{lm} is the surface area. The inviscid fluxes are calculated based on the flux difference splitting of Roe

$$F_{lm} = \frac{1}{2}\big(\bar{u}(U_l + U_r) - |\bar{u}|(U_r - U_l)\big) \ , \tag{3.34}$$

where U is the vector of the unknown variables, consisting of the velocity and the void fraction. The indexes l and r denote the left and right sides of the face, respectively. The cell face pressures are calculated according to

$$p_{lm,1} = \frac{h^+\rho_m p_l + h^-\rho_l p_m}{h^+\rho_m + h^-\rho_l} \ , \tag{3.35}$$

due to the possible discontinuity in the density (cf. Sec. 3.2.6).

The convection velocity $\bar{u}$ is calculated as a weighted average of the left

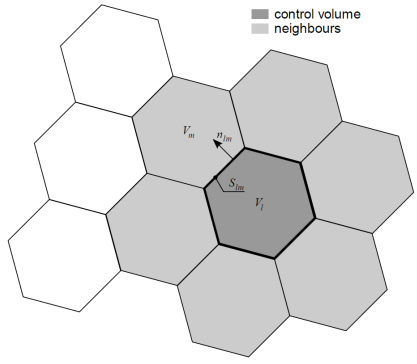


Figure 3.4: Finite volume discretization used in YAFFA (Mikkola, 2009). Note that in this work, rectangular control volumes are used.

Table 3.1: The Gamma Differencing Scheme (GDS) of Jasak (1996).

Value in cell l	**Value at face** lm	**Scheme**
$\tilde{\alpha}_l \leq 0$	$\tilde{\alpha}_l$	First-order upwind
$0 < \tilde{\alpha}_l \leq \beta_m$	$-\frac{\tilde{\alpha}_l^2}{2\beta_m} + \left(1 + \frac{1}{2\beta_m}\right)\tilde{\alpha}_l$	Blending
$\beta_m \leq \tilde{\alpha}_l < 1$	$\frac{1}{2} + \frac{1}{2}\tilde{\alpha}_l$	Central difference
$\tilde{\alpha}_l \geq 1$	$\tilde{\alpha}_l$	First-order upwind

and right side cells of the face. The weighing factor depends on the normalized distance between the opposite cell centres from the face. The discrete counterpart of the convected velocity components $\vec{V}$ on the cell face is based on the first-order Taylor expansion in a control volume l. The cell face value is

$$u_{i,lm} = u_{i,l} + \gamma \frac{\partial u_{i,l}}{\partial x_j} \Delta x_j \ , \tag{3.36}$$

where the index i denotes the i:th component of the vector, Δx_j is the distance between the cell face lm and its center l; γ is a global user-defined parameter controlling the amount of the correction; summation is implied over j. A Rhie-Chow type damping term is added to the convection velocity in calculation of the error in mass balance; this is discussed in Sec. 3.2.5.

Two different extrapolation procedures have been implemented for the cell face value of the void fraction. These are the Gamma Differencing Scheme (GDS) of Jasak (1996) and the Inter-Gamma Differencing Scheme (IGDS) of Jasak and Weller (1995), which are based on the normalized variable (NV) formulation. It is here reminded that the NV formalism changes

Table 3.2: The Inter-Gamma Differencing Scheme (IGDS) of Jasak and Weller (1995).

Value in cell l	**Value at face** lm	**Scheme**
$\tilde{\alpha}_l \leq 0$	$\tilde{\alpha}_l$	First-order upwind
$0 < \tilde{\alpha}_l \leq \frac{1}{2}$	$-2\tilde{\alpha}_l^2 + 3\tilde{\alpha}_l$	Blending
$\frac{1}{2} < \tilde{\alpha}_l < 1$	1	First-order downwind
$\tilde{\alpha}_l \geq 1$	$\tilde{\alpha}_l$	First-order upwind

in the case of arbitrary computational grids, and this obliges one to use a specific blending function, usually denoted as f_x, in the expression of the non-normalized quantity on the cell face (Jasak, 1996; Jasak et al., 1999). Here only orthogonal, or very closely orthogonal, grids are used, so further attention is not needed for this. The GDS is defined in Tab. 3.1, where β_m is the pre-specified constant for the scheme, and the IGDS is given in Tab. 3.2. More detailed discussion of these schemes and the normalized variable ($\tilde{\alpha}$) formulation follows in Sec. 3.3. In this study, the IGDS is used in computations with Yaffa.

Finflo The inviscid fluxes on the cell faces are based on the Roe's method

$$F_{i+1/2} = \frac{1}{2}(F(U_l) + F(U_r)) - \frac{1}{2}|\tilde{A}|(U_r - U_l) \ , \tag{3.37}$$

where F is the flux vector in x-direction, as in Eq. (3.13) consisting of convection and convected velocities as well as pressures, total energy and turbulence variables. $\tilde{A}$ is the Jacobian of the flux vector that is based on the Roe average. That is, it is based on the familiar expression $\partial F/\partial U$ but satisfies certain conditions such that the solution of the linear problem becomes an approximate solution of the Riemann problem. These methods contain intrinsic dissipation that is due to the upwind-biased extrapolation (Siikonen, 2011; Tannehill et al., 1997). Here, only the different velocity components are discussed together with the convective void fractions. The cell face pressures are obtained using simple averages of the cells surrounding the face $i + 1/2$. The convected turbulence quantities are calculated with a second-order upwind extrapolation.

The convective values of $\vec{V}$ and α are MUSCL-extrapolated to the cell face (cf. Sec. 3.3.2). In the calculation of the mass fluxes $\dot{m}_k = S\rho_k\alpha_k\bar{u}$ the values of α on the cell faces are always extrapolated using the second-order

upwind method and limited using the van Albada limiter; the concept of a limiter is introduced to allow the TVD propery. This will be discussed later in Sec. 3.3.2. Additionally, the void fractions are constrained to the range $[0, 1]$. In this work, another limiter, the SUPERBEE of Roe, was implemented in the FINFLO code. The case in Sec. 4.2 was calculated using both the original van Albada limiter and the SUPERBEE limiter. The motivation for this is evident from Sec. 3.3. The van Albada limiter is

$$\phi_{va} = \frac{r + r^2}{1 + r^2} \ , \tag{3.38}$$

and the SUPERBEE is

$$\phi_{sb} = \mathsf{max}\big(0, \mathsf{min}(2r, 1), \mathsf{min}(r, 2)\big) \ , \tag{3.39}$$

where r is the ratio of upwind slopes to the downwind ones at the cell face. These will be detailed later in Sec. 3.3.2. The MUSCL-formula is implemented in an alternative way

$$U^l_{i+1/2} = U_i + \overbrace{\frac{\phi(r)}{4}}^{\texttt{PHIBI}} \big(\kappa_1(U_i - U_{i-1}) + \kappa_2(U_{i+1} - U_i)\big) \quad \text{and} \tag{3.40}$$

$$U^r_{i+1/2} = U_{i+1} - \underbrace{\frac{\phi(r)}{4}}_{\texttt{PHIFI}} \big(\kappa_2(U_{i+1} - U_i) + \kappa_1(U_{i+2} - U_{i+1})\big) \ , \tag{3.41}$$

where the corresponding slope is used in the argument of the limiter function ϕ, and the superscripts denote the left (l) or right (r) side of the face. `PHIBI` and `PHIFI` are the coefficients in the code. In FINFLO, the void fraction is extrapolated in subroutine `DXTRAP`. Three variables are needed for the limiter function

$$\texttt{DIFF0} = \texttt{U(IT1)} - \texttt{U(IT0)} \ , \tag{3.42}$$

$$\texttt{DIFF1} = \texttt{U(I)} - \texttt{U(IT1)} \quad \text{and} \tag{3.43}$$

$$\texttt{DIFF2} = \texttt{U(IT2)} - \texttt{U(I)} \ , \tag{3.44}$$

where the array pointers are seen in Fig. 3.5. The differences correspond to those used in the MUSCL extrapolations above. Note that for the void fraction $\kappa_2 = 0$. The SUPERBEE limiter is then implemented into the code

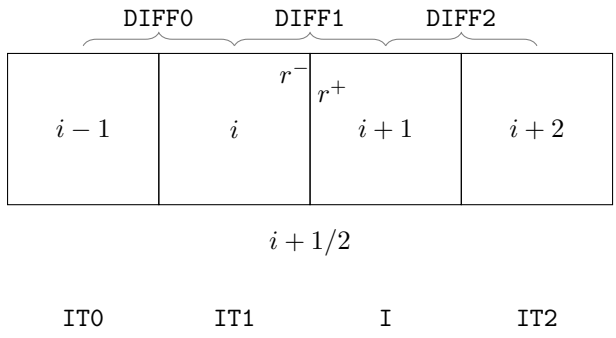


Figure 3.5: Definitions of some array pointers and variables used in the code.

simply

$$\texttt{PHIBI} = \texttt{MAX}(0.0, \texttt{MIN}(2.0 * \texttt{RIII}, 1.0), \texttt{MIN}(\texttt{RIII}, 2.0)) \quad \text{and} \tag{3.45}$$

$$\texttt{PHIFI} = \texttt{MAX}(0.0, \texttt{MIN}(2.0 * \texttt{RII2}, 1.0), \texttt{MIN}(\texttt{RII2}, 2.0)) \ , \tag{3.46}$$

where the expressions `MAX` and `MIN` correspond to the Fortran statements of a maximum or minimum of the real arguments. The variables `RIII` and `RII2` are

$$\texttt{RIII} = \texttt{DIFF0/DIFF1} \quad \text{and} \tag{3.47}$$

$$\texttt{RII2} = \texttt{DIFF2/DIFF1} \ , \tag{3.48}$$

that is, they are the different sided slope ratios r^- and r^+, respectively.

In an incompressible case, the convection velocities need additional dissipation terms that couple them with the pressures and hence avoids the possible formation of checkerboard-like pressure fields (Patankar, 1980; Siikonen, 2011, 2013). This dissipation is added to the mass flux formula in the so-called spirit of the Rhie and Chow interpolation. The convection velocity is calculated as

$$\bar{u}_{i+1/2} = \frac{1}{2}(\bar{u}_{i+1} + \bar{u}_i) + C\frac{S_{i+1/2}}{\mathsf{A}_{P,i+1/2}}(p_{i+2} - 3p_{i+1} + 3p_i - p_{i-1}) \ , \tag{3.49}$$

that is, the convection velocities at the cell centres surrounding the face $i + 1/2$ together with the neighbouring pressure values are used to calculate the convective velocity at the cell face. The term A_P comes from the lin-

earizations of the momentum equations (see Sec. 3.2.1). The factor C can be used to reduce the added dissipation. The above term is of the fourth order, and the resulting truncation error $\mathcal{O}(\Delta x^2)$. (Siikonen, 2011)

3.2.3 Diffusion term

Yaffa In this study, the Euler equations were used in computations with YAFFA.

Finflo In this work, the viscous fluxes are evaluated based on the thin layer approximation. In the code, the thin layer model can be activated in any coordinate direction. This approximation posits on the fact that in the boundary layer, the gradient of a variable perpendicular to the wall is clearly larger than that along the wall. The approximation is also valid if the cell face areas are much greater in the direction along the wall than they are perpendicular to it. The diffusion fluxes in the momentum equations include derivatives on the cell face. The calculation of the velocity derivative is, for example, in the x-direction in the cell i

$$\left.\frac{\partial u}{\partial x}\right|_i \approx \frac{(n_x u)_{i+1/2} - (n_x u)_{i-1/2}}{d_i} \ , \tag{3.50}$$

where d_i is the thickness of the cell. This formula is applied to a shifted control volume $V_{i+1/2}$, and used to replace the derivatives in of the viscous fluxes in Eq. (3.15) yielding

$$F_{v_{2,i+1/2}} = -\frac{\mu_{eff}}{d_{i+1/2}}\big((u_{i+1} - u_i) + \frac{n_{x,i+1/2}}{3}(\bar{u}_{i+1} - \bar{u}_i)\big) \ , \tag{3.51}$$

where $\mu_{eff} = \mu + \mu_t$ and the contravariant velocity defined now as

$$\bar{u}_{i+1} - \bar{u}_i = n_{x,i+1/2}(u_{i+1} - u_i) + n_{y,i+1/2}(v_{i+1} - v_i) + n_{z,i+1/2}(w_{i+1} - w_i) \ . \tag{3.52}$$

Equations similar to Eq. (3.51) are used for diffusion fluxes in y and z-directions with the corresponding velocity components used in the first term on the right hand side.

3.2.4 Source term

The source term contains non-zero components only for the momentum equations, and for turbulence closure. The acceleration due to gravity is a body force that applies to the whole fluid in a cell. Its discrete representation was $\mathcal{V}_i\vec{Q}_i$ where the source term is now $\rho\vec{g}$. The source term has not been linearized *per sé*, but the value for density is obtained from the previous cycle. Presumably this does not affect the stability.

Yaffa Recalling that the code solves the Euler equations in primitive form, the source term contribution changes to

$$\vec{Q}_l = \vec{g} \; . \tag{3.53}$$

This manipulation eliminates the large term from the momentum equations.

Finflo In the incompressible solution of FINFLO pressure differences are used. The source term is then (Miettinen et al., 2006)

$$\vec{Q}_i = (\rho_i - \rho_\infty)\vec{g} \; , \tag{3.54}$$

where ρ_∞ is the free stream (reference) density. If the density of either of the phases in question is used as the free stream density, this term reduces to zero for that phase. Additionally, the source term is linearized in a peculiar way to $\rho^{n+1} = \rho_0$, that is, setting this term to a constant value throughout the iteration by

$$\rho(x,y,z)^{n+1} = \begin{cases} \rho_g & \text{if } z > z_0 \\ \rho_l & \text{otherwise,} \end{cases} \tag{3.55}$$

where z_0 denotes the initial still water level. The turbulence model is presented in Finflo (2013)

3.2.5 Pressure correction stage

As was already discussed in Sec. 2.3 the conservation of mass within the iterative solution is of vast importance. This is enforced mainly through the

pressure correction stage. One could note that throughout the Sec. 3.2.1 the enforcement of the mere momentum balance, Eq. (3.2), was pursued. This by no means states that mass balance, Eq. (3.1), is reached in the approximative solution since it was not explicitly sought at any stage above; rather, it was implicitly assumed in the coefficients of the linearized momentum equation. Furthermore, it provided no description for the pressure field p. The pressure gradient appears in the Navier-Stokes equations (2.2) and the pressures are explicitly needed in the fluxes (*e.g.* Eqs. (3.10) and (3.13)) at the cell faces. Since most flow situations within the problems encountered at naval hydrodynamics can be considered as incompressible, there is no equation of state for the pressure. To overcome these difficulties, one general way in computational fluid dynamics (cf. Patankar (1980)) is to write instead for the new time level

$$u_i^{n+1} = u_i^* + u_i' \ , \tag{3.56}$$

where the latter part on the right hand side is so-called velocity correction, and i denotes any of the components. Now u_i^* is an intermediate velocity field that satisfies the momentum equation, but it needs to be corrected by the amount of u' in order for it to satisfy the mass balance as well. Similar decomposition is possible for the pressure

$$p^{n+1} = p^* + p' \ , \tag{3.57}$$

where the p^* is a guessed pressure field for the solution of the momentum equations, *i.e.* it follows from the initial pressure field or is that of the previous iteration cycle. p' the correction part to provide a pressure field that promotes the mass balance as well. (Previously the primed values were used to denote the presence of turbulent fluctuation; *per sé*, they are not connected to the pressure correction stage). The mass balance for the new time level is

$$\sum_j \dot{m}_j^{n+1} = \sum_j (S\rho\bar{u})_j^{n+1} = 0 \ , \tag{3.58}$$

where $\bar{u}_j = (\vec{V} \cdot \vec{n})_j$ are the convection velocities at the cell faces j. Substituting Eq. (3.56) to the mass balance gives

$$\sum_j \dot{m}_j^* = -\sum_j \dot{m}_j' \ , \tag{3.59}$$

where the summation over j denotes that over the faces of the computational cell. The likely mass imbalance for a cell i, $\Delta\dot{m}_i^*$, resulting from the velocity field that satisfies the momentum equations is the left hand side of Eq. (3.59). The pressure correction equation is obtained by replacing the contravariant velocity corrections $\bar{u}'$ in the mass flux correction $\dot{m}' = S\rho\bar{u}'$ by an expression involving the pressure corrections p' at the cell faces. The link between these pressure corrections and the velocity corrections is found through the linearized momentum equation (3.31). The linearization of the continuity equation can be done in many different ways, and this will be described later.

The pressure correction equation is based on the expression (3.59). In other words, if the mass balance is satisfied, the pressure corrections are zero, *i.e.* $p' = 0$. Thus, the nature of the pressure correction stage is essentially in providing guidance to the solution so that it converges to the mass balance as well. This allows for some simplifications in its derivation that could lead to faster convergence or a more stable algorithm. Earlier the kinematic boundary condition was directly linked to the same aspect (the mass balance). This implicit coupling of α and p is then further strengthened by the dynamic boundary condition.

The values u^* and p^* already satisfy the momentum equations. The substitution of the decompositions (3.56) and (3.57) to Eq. (3.31) applied to a shifted control volume, supplied with apt exploitation of the aforementioned facts gives a link

$$\mathsf{A}_{P,i+1/2}u'_{i+1/2} = -S_{i+1/2}n_{x,i+1/2}(p'_{i+1} - p'_i) \ , \tag{3.60}$$

where $u'_{i+1/2}$ are the u-velocity corrections at the cell face $i + 1/2$, $S_{i+1/2}$ the cell face area and $n_{x,i+1/2}$ the x-component of its normal vector. The term $\mathsf{A}_{P,i+1/2}$ is based on the linearizations of the flux vectors, *i.e.* they are the coefficients of the momentum equations for their implicit stage, Eq. (3.30). Again, similar forms are obtained for y and z-directions as well. Eq. (3.60) is the SIMPLE approximation commonly taken in computational fluid dynamics. (Ferziger and Perić, 1999; Patankar, 1980; Siikonen, 2013; Tannehill et al., 1997)

After solving the pressure correction equation, which will be given below for both codes, the mass balance is exactly satisfied. The corrections on the

velocities is performed using Eq. (3.60). Conversely, the momentum balance was not explicitly enforced within this step. The procedure is iterative, and a converged solution is reached in several cycles where these steps are repeated. A known feature of the SIMPLE algorithm is the need for an under-relaxation of the pressure corrections, *i.e.* the pressures are updated from

$$p^{n+1} = p^* + \alpha_p p' \ , \tag{3.61}$$

where $\alpha_p \leq 1$ is the under-relaxation factor for the pressure. The SIMPLE procedure may overestimate the pressure correction due to the approximation (3.60), which in turn can lead to unstable computations. For this reason, it may be necessary to apply only a fraction of p' to the pressure. The values for α_p depend on the case at hand. Patankar (1980) recommends the range $\alpha_p \in [0.5, 0.8]$ but quite usually significantly smaller values are the only cure for the calculations to converge. In addition to the pressures, a similar practice can be applied to the velocity corrections via $u^{n+1} = u^* + \alpha_u u'$. If this is done, the resulting velocity field no longer satisfies the continuity condition exactly. Such under-relaxation is unfortunately frequently necessary for the same reasons as above. The problem may be so non-linear that the iterative algorithm must be heavily under-relaxed; it does not affect the converged solution. It only may take even several tens of thousands of iteration cycles to reach it.

The linearization of the continuity equation is another aspect of the pressure correction methods. It can be done in many different ways depending on the problem at hand and the flow equations being solved. Note that Eq. (3.59) does not yet contain indication of the two-phase problem, but is based on the traditional way to linearize the equation since such is the current practice. In principle, in this case (cf. Sec. 2.2), the pressure correction equation can be based on the volumetric fluxes as well, *i.e*, on the incompressibility constraint $\sum_j S_j \bar{u}_j = \sum_j \dot{\mathcal{V}}_j = 0$. This form guarantees conservative volumetric fluxes, rather than mass fluxes, which can in fact be argued to be advisable in the VOF algorithm. The pressure correction stage of YAFFA is based on the balance of the volumetric fluxes, whereas in FINFLO, the starting point is the phasic mass balance. It is important to remember the principle of the guiding role of the whole procedure. The pressure correction equation is formed differently in the codes used here and will be discussed below.

Yaffa The pressure correction equation applied in YAFFA is based on the volumetric conservation. In this work, the SIMPLE method is applied, or the pressure correction equation follows from the discretized and simplified momentum equation. The form of that equation differs slightly in unstructured discretizations,

$$A_l u'_{i,l} = -\sum_{lm} S_{lm} \hat{p}'_{lm} n_{i,lm} \ , \tag{3.62}$$

where the coefficients A_l follow from the upwind linearization, $u'_{i,l}$ are the components of the velocity corrections for the cell l, S_{lm} are the cell face areas, $\hat{p}'_{lm}$ are the pressure corrections on the cell faces and $n_{i,lm}$ are the components (i) of the normal vector of the cell face. This has the same meaning as the expression (3.60), but with the exceptions that it is formulated directly for the cell l, and the terms $\hat{p}'_{lm}$ do not have the dimensions of pressure. The latter is due to the primitive form of the momentum equations that are being solved; also the diagonal term changes to

$$A_l = \frac{\mathcal{V}_l}{\Delta t} - \sum_m a_{lm} \ . \tag{3.63}$$

This follows from the linearization of the momentum equations in a similar fashion as was done previously in Sec. 3.2.1 with FINFLO. The off-diagonal terms are now

$$a_{lm} = -S_{lm} \mathsf{max}(-\bar{u}_{lm}, 0) \ . \tag{3.64}$$

Note that the matrix consists of volumetric fluxes, rather than mass fluxes. The convection velocity correction on the cell faces are needed for the continuity equation, and these are obtained by projecting the Cartesian components to the cell face via a dot product of the Eq. (3.62), divided by the diagonal coefficient, and the normal vector of the face, and by using suitable averages. The right hand side can be then transformed using the Gauss theorem, and the correction of the convection velocity is

$$\bar{u}'_{lm} = -\frac{\mathcal{V}_{lm}}{A_{lm}} \left(\frac{\partial \hat{p}'}{\partial n} \right)_{lm} \ . \tag{3.65}$$

The cell face volumes and inverses of the coefficients A_{lm} are taken as the average values of the cells l and m surrounding the face. The pressure gradient is approximated with the difference of the values at these cells, divided by the distance of their centres in the normal direction to the face.

The convection velocity is then

$$\bar{u}'_{lm} = \frac{\mathcal{V}_l + \mathcal{V}_m}{A_l + A_m} \cdot \frac{\hat{p}'_m - \hat{p}'_l}{(x_{i,m} - x_{i,l})n_{i,lm}} \ , \tag{3.66}$$

where summation is implied over the vector components i. The mass balance reduces into a volumetric one, and is calculated in the spirit of the Rhie-Chow interpolation basing on the expression

$$\dot{\mathcal{V}}^*_{lm} = S_{lm} \left(\bar{u}^*_{lm} + C \frac{\mathcal{V}_{lm}}{A_{lm}} d_{lm} \right) \ , \tag{3.67}$$

where $\dot{\mathcal{V}}^*$ is the volumetric flux on the cell face, C is the Davidson factor used to control the amount of the damping, and the term d_{lm} is originally used in Rhie-Chow -type expression in relating the convection velocity explicitly to the pressures and pressure gradients surrounding the face. The Rhie-Chow interpolation provides an explicit link between the cell face velocities and pressures, and it is only needed in calculation of the error in the volumetric (or mass) flux balances, Eq. (3.67). The first pressure gradient in this interpolation should correspond to the discretization used for the temporary velocity $\bar{u}^*_{lm}$, that is, in the solution of the momentum equations. The second gradient, that is subtracted from the first one, is discretized identical to that used in the normal derivative of the pressure corrections. Because of the two-phase problem (cf. Sec. 3.2.6), the term d_{lm} is based on different expressions

$$d_{lm} = \left(\frac{\partial \hat{p}}{\partial n} \right)_{lm,1} - \left(\frac{\partial \hat{p}}{\partial n} \right)_{lm,2} \tag{3.68}$$

for these pressure gradients. For the first gradient, the pressure on the cell face is calculated similarly as in the inviscid fluxes from

$$p_{lm,1} = \frac{h^+ \rho_m p_l + h^- \rho_l p_m}{h^+ \rho_m + h^- \rho_l} \ , \tag{3.69}$$

where the distances $h^\pm$ and cells l and m are shown in Fig. 3.6. This equation stems from the work of Queutey and Visonneau (2007). The cell face pressures are multiplied by the normal to the face, and then the Gauss theorem is used to transform these into gradients. In the code, these gradients are calculated to the cell centres, and divided by the values of the respective densities. The interpolation of these gradients to the cell faces is done using a distance weighed formula. The second gradient is calculated from the

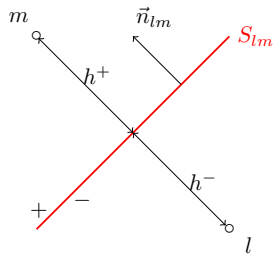


Figure 3.6: Notations around the cell face lm.

expression

$$\left(\frac{\partial\hat{\rho}}{\partial n}\right)_{lm,2} = \left(\frac{1}{\hat{\rho}}\frac{p_m - p_l}{h}\right)_{lm} , \tag{3.70}$$

where the distance $h = h^+ + h^-$, and the distance weighed density $\hat{\rho}_{lm}$ is defined by

$$\hat{\rho}_{lm} = \frac{h_l\rho_l + h_m\rho_m}{h} . \tag{3.71}$$

An intermediate stage is in order by substituting the correction of the convection velocity, Eq. (3.66), to the incompressibility constraint $\sum_j \dot{\mathcal{V}}'_j = -\sum_j \dot{\mathcal{V}}^*_j$ to yield

$$\sum_m S_{lm}\left(\frac{\mathcal{V}_l + \mathcal{V}_m}{A_l + A_m}\frac{\hat{p}'_m - \hat{p}'_l}{(x_{i,m} - x_{i,l})n_{i,lm}}\right) = -\sum_{lm}\dot{\mathcal{V}}^*_{lm} , \tag{3.72}$$

whence the final pressure correction equation is available

$$a_{ll}\hat{p}'_l + \sum_m a_{lm}\hat{p}'_m = -\sum_{lm}\dot{\mathcal{V}}^*_{lm} , \tag{3.73}$$

where the diagonal term is the sum of the off-diagonal ones

$$a_{ll} = -\sum_m a_{lm} , \tag{3.74}$$

and those are

$$a_{lm} = -\frac{V_l + V_m}{A_l + A_m}\frac{S_{lm}}{(x_{i,m} - x_{i,l})\, n_{i,lm}} . \tag{3.75}$$

The solution of the system of equations, Eq. (3.73), yields the pressure corrections, but normalized with ρ. Hence, they must be multiplied with the density when the pressures are corrected, or $p^{n+1} = p^* + \alpha_p\rho^n\hat{p}'$. Here the previous value of the density is used, since the void fraction equation is solved

after the pressure correction stage as in Fig. 3.2. The system is solved using the conjugate gradient (CG) algorithm.

Finflo The pressure correction method applied in FINFLO belongs to the coupled approaches, since now the residuals of mass and momentum are calculated simultaneously and only once during an iteration cycle. The coupled set of equations were avoided by manipulating the explicit residuals via Eq. (3.18). The continuity equations form the basis of the linearization for the mass balance. The linearization resides in separating the change in density into suitable expressions such as

$$\Delta\rho_k = \frac{\partial\rho_k}{\partial p}p' + \frac{\partial\rho_k}{\partial h_k}\Delta h_k \ , \tag{3.76}$$

where h_k is the enthalpy for phase k. That is, the time derivative is not dropped despite ρ_k is constant, since the pressure correction equation results in a very stiff system, cf. Eq. (3.73). Traditionally, the coefficients of the pressure correction equation come directly from the correction term in mass balance into which the link of Eq. (3.60) is substituted. One writes instead

$$\mathcal{V}_i\frac{\Delta\rho}{\Delta t} + \sum_j \left(S\rho\bar{u}'\right)_j = -\Delta\dot{m}_i \tag{3.77}$$

for the mass flux correction for the cell i. Slight diagonal dominance is enforced with the time derivative term. $\Delta\rho$ can be interpreted as an imbalance in the mass flows at the time level n (Siikonen, 2011). Now it is expanded approximatively for phase k

$$\begin{aligned}
&\rho_{k,i}\mathcal{V}_i\frac{\Delta\alpha_{k,i}}{\Delta t} + \sum_{out} S_j\rho_{k,j}\bar{u}_j\Delta\alpha_{k,i} + \\
&\left(\frac{\mathcal{V}_i\alpha_{k,i}}{\Delta t} + \sum_{out} S_j\bar{u}_j\alpha_{k,j}\right)\frac{\partial\rho_k}{\partial p}p_i' + \\
&\left(\frac{\mathcal{V}_i\alpha_{k,i}}{\Delta t} + \sum_{out} S_j\bar{u}_j\alpha_{k,j}\right)\frac{\partial\rho_k}{\partial h_k}\Delta h_{k,i} \ .
\end{aligned}$$

Further simplifications follow from ignoring the outgoing sums in the first two terms and disregarding the linearizations with respect to enthalpy. An approximation for the speed of sound can be identified, or $1/c_k^2 \approx \partial\rho_k/\partial p$.

Taking these into account leaves

$$\rho_{k,i}\mathcal{V}_i\frac{\Delta\alpha_{k,i}}{\Delta t}+\frac{\mathcal{V}_i\alpha_{k,i}}{\Delta t c_{k,i}^2}p_i'+\sum_j\left(S\alpha_k\rho_k\bar{u}'\right)_j=-\Delta\dot{m}_{k,i}\ . \tag{3.78}$$

The error in mass balance for the cell i after the solution of the momentum equations for the mixture (from the previous iteration cycle) is the sum of Eq. (3.78) over both phases. A term resembling the speed of sound in the mixture is defined as

$$\frac{1}{\rho c_0^2}=\frac{\alpha}{\rho_g c_g^2}+\frac{1-\alpha}{\rho_l c_l^2}\ , \tag{3.79}$$

where c_g is the speed of sound in the gas phase and c_l the speed of sound in the liquid phase. After division by the phase density ρ_k the changes in the void fraction can be eliminated on the basis of $\Delta\alpha_g+\Delta(1-\alpha_g)=0$, and by writing that sum one obtains

$$\frac{\mathcal{V}_i}{\rho_i c_{0,i}^2\Delta t}p_i'+\sum_k\frac{1}{\rho_{k,i}}\sum_j(S_j\alpha_{k,j}\rho_{k,j}\bar{u}_j')=-\sum_k\frac{\Delta\dot{m}_{k,i}}{\rho_{k,i}}\ . \tag{3.80}$$

Using the first-order linearization for the cell face densities $\alpha_{k,j}\rho_{k,j}$ in case of an outflow, and further approximating them with that of the cell i even in the case of an inflow, it is possible to eliminate them. The void fraction may differ significantly on the different sides of the face, and consideration could be put on it; now it is disregarded. The correction of the convection velocity is obtained by taking the dot product with the Cartesian components, Eq. (3.60), and the normal vector of the cell faces. Substituting this to the above expression results in

$$\frac{\mathcal{V}_i}{\rho_i c_{0,i}^2\Delta t}p_i'-\sum_j\frac{S_j^2}{\bar{\mathsf{A}}_{P,j}}(p_{j+}'-p_i')=-\sum_k\frac{\Delta\dot{m}_{k,i}}{\rho_{k,i}}\ , \tag{3.81}$$

where the index $j+$ denotes the value in the cell on the other side of the face j, the other neighbour is always the cell i. The error in mass balance for phase k is weighed with the density of that phase, and their sum is on the right hand side. The coefficients $\mathsf{A}_{P,j}$ are the same for all momentum equations since they are not updated after the solutions of the consecutive components. It is important to calculate the coefficient correctly at the cell

face. A robust way of doing that is

$$\frac{1}{\bar{\mathsf{A}}_{P,i+1/2}} = \frac{1}{2}\left(\frac{1}{\mathsf{A}_{P,i+1}} + \frac{1}{\mathsf{A}_{P,i}}\right) \ , \tag{3.82}$$

since use of a simpler average would result in a very unstable algorithm. Using similar compass notation as in Sec. 3.2.1 the two-dimensional pressure correction equation can be written as

$$\begin{aligned}
\mathsf{A}_W &= -\left(\frac{S^2}{\bar{\mathsf{A}}_P}\right)_{i-1/2} \\
\mathsf{A}_E &= -\left(\frac{S^2}{\bar{\mathsf{A}}_P}\right)_{i+1/2} \\
\mathsf{A}_S &= -\left(\frac{S^2}{\bar{\mathsf{A}}_P}\right)_{j-1/2} \\
\mathsf{A}_N &= -\left(\frac{S^2}{\bar{\mathsf{A}}_P}\right)_{j+1/2} \\
\mathsf{A}_P^p &= \frac{\mathcal{V}_i}{\rho_i \Delta t c_{0,i}^2} - \mathsf{A}_W - \mathsf{A}_E - \mathsf{A}_S - \mathsf{A}_N \ ,
\end{aligned} \tag{3.83}$$

where the term $\bar{\mathsf{A}}_{P,i+1/2}$ denotes the diagonal term of the linearized momentum equations (3.31), averaged to the cell face. The use of Eq. (3.81) for the solution of the pressure corrections p' has proven robust and efficient for the cases that did not involve mass transfer; the very desirable feature within this work as well. This discussion will be resumed in Chap. 5.

3.2.6 Two-phase methodology

Yaffa The inclusion of the segregated phases is done with the standard convection equation of the void fraction α, Eq. (2.20). However, in order to remove the steep density gradient from the hydrostatic pressure contribution, the non-conservative form of the momentum equations is applied. In addition, the bulk flow solution procedure is modified to account for the discontinuity in the material properties (Hänninen and Mikkola, 2007). Some of these changes correspond to those introduced by Queutey and Visonneau (2007). In order to account for the jump conditions along the interface of the two-phases, the cell faces pressures are interpolated in the inviscid fluxes

according to

$$p_{lm} = \frac{h^+ \rho_m p_l + h^- \rho_l p_m}{h^+ \rho_m + h^- \rho_l} \ , \tag{3.84}$$

where h denotes the distance between the cell centre and the face, and the indexes m and l denote the cells on both sides of the face (Fig. 3.6). The use of simple distance weighted averages leads to an inaccurate approximation of the cell face pressure profile (cf. Sec. 2.3), whereas Eq. (3.84) was specifically designed to account for the jump conditions. Queutey and Visonneau (2007) showed that these types of interpolations satisfy the jump conditions due to discontinuity in the density in the hydrostatic case. Similarly, the pressure gradient on the cell face is normalized by a modified density for the pressure correction stage in the Rhie-Chow interpolation

$$\left(\frac{\vec{\nabla} p \cdot \vec{n}}{\rho} \right)_{lm} = \frac{1}{\hat{\rho}_{lm}} \frac{p_m - p_l}{h} \tag{3.85}$$

with the distance weighted density defined by

$$\hat{\rho}_{lm} = \frac{h_m \rho_m + h_l \rho_l}{h} \ . \tag{3.86}$$

This was described in Sec. 3.2.5. The pressure gradient normalized by the density is continuous through the face (Queutey and Visonneau, 2007).

The inclusion of the two-phase methodology in the solution algorithm consists of two separate loops within an iteration cycle or time step. This was illustrated in Fig. 3.2. The bulk flow equations are solved within an inner loop, and the convection equation of the void fraction is solved in an outer loop. The solution of the void fraction is based on the same procedure as the momentum equations, that is, the loop begins with calculation of the explicit residuals of α through the flux balance $\alpha\bar{u}$, which forms the right hand side of the implicit stage. The implicit stage yields the change $\Delta\alpha$. After updating the void fraction values via $\alpha^{n+1} = \alpha^n + \Delta\alpha$, the iteration is advanced to the next step. These loops can be repeated if required, and typically $\mathcal{O}(20)$ is sufficient. The solution procedure of Yaffa was already given in Sec. 3.2.1, and a similar algorithm to solve the time evolution of the void fraction is detailed in Sec. 3.3.

Additionally, the governing equations in Yaffa were modified into the primitive form. This eliminates the need of calculating the density values

on the cell faces. It was observed in development of the interface capturing model that the interpolation of the coefficients like A_l on the cell faces posed a problem when the density discontinuity was present. This mainly deals with the pressure correction stage, and was discussed earlier in the respective section.

Finflo The void fractions α and the mass fractions x are used interchangeably in the solution procedure. The void fractions are used in the explicit stage, whereas the mass fraction is used in the implicit stage. The continuity equation for the gas phase is utilized. This is rewritten for convenience

$$\mathcal{V}_i \frac{\Delta(\alpha_i \rho_{g,i})}{\Delta t} = - \sum_{faces} S_j \alpha_j \rho_{g,j} \bar{u}_j \ . \tag{3.87}$$

The MUSCL-discretization is used for the convective fluxes of the void fraction. The values are extrapolated from the centre points of the control volumes to the faces using the second-order upwind-biased method, and the fluxes are limited with either the van Albada or SUPERBEE limiter. The explicit residual is the difference in the fluxes into a computational cell i through its faces j, or the right hand side of Eq. (3.87). The residual is $\Delta(\alpha_i \rho_{g,i})$.

For the implicit stage, the convective fluxes can be linearized in an upwind manner as

$$(\dot{m} x_g)^{n+1}_{i+1/2} = (\dot{m} x_g)^{n}_{i+1/2} + \dot{m}_{i+1/2} \Delta x_{g,i} \tag{3.88}$$

if flow is in the direction of the increasing index i. Since the mixture density was used, the mixture mass flow appears in the expression and consequently exactly the same routines that are used for the change of the velocity components can be used for the solution of the change of the gas mass fraction. Note that earlier in Sec. 2.2 and also later in Sec. 3.3 the convection equation for the void fraction is assessed. One should remember that based on the assumption of incompressibility, the individual field densities could be eliminated from the continuity equation, and the algorithm would result to that consisting merely of α and the contravariant velocity; that is the *de facto* standard in interface capturing-type methods employing the VOF concept. Now also the novel idea of Miettinen et al. (2006), essentially Eq. (3.18), can be used. For illustration, insertion of the above linearizations to

a two-dimensional continuity equation for the gas mass fraction yields

$$\begin{aligned}\mathcal{V}_i \frac{d(\rho x_g)_i}{dt} &= \underbrace{-F_{i+1/2} + F_{i-1/2} - F_{j+1/2} + F_{j-1/2}}_{=R_i} \\ &- \dot{m}_{i+1/2}\Delta x_{g,i} - \dot{m}_{j+1/2}\Delta x_{g,i} \\ &+ \dot{m}_{i-1/2}\Delta x_{g,i-1} + \dot{m}_{j-1/2}\Delta x_{g,i-imax} \ ,\end{aligned} \tag{3.89}$$

where R_i is the explicit residual. There the discretization practices taken for α (the void fraction was used in the explicit stage) on the cell faces are of utmost importance. These are discussed in Sec. 3.2.2, and further insight is shed in Sec. 3.3.2. The change in the mass fraction is also transformed

$$\rho\Delta x_k = \Delta\big(\rho x_k\big) - x_k\Delta\rho \ . \tag{3.90}$$

The term $\Delta\big(\rho x_k\big)$ is the error in mass balance for phase k. For the term on the left hand side of Eq. (3.89) can be simplified from linearization of the continuity equation

$$\mathcal{V}_i \frac{\Delta\rho_i}{\Delta t} = \overbrace{-\sum_j \dot{m}_j^n}^{=R_i^m} - \sum_j \delta\dot{m}_j \ , \tag{3.91}$$

where $\Delta\rho$ is interpreted as the mass imbalance from the previous iteration, the left hand side can then be written as

$$\mathcal{V}_i\rho_i \frac{\Delta x_i}{\Delta t} + x_i \left(R_i^m - \sum_j \delta\dot{m}_j \right) \ .$$

If the mass balance is assumed to hold implicitly, the correction term $\sum_j \delta\dot{m}_j$ can be dropped. The terms involving the mass flows can then be grouped, and the implicit stage for the gas mass fraction is analogously

$$\left(\frac{\mathcal{V}_i\rho_i}{\Delta t} - \sum_{in} \dot{m}_j \right) \Delta x_i + \sum_{in} \dot{m}_j \Delta x_j = R_i - x_i R_i^m \ , \tag{3.92}$$

where the summation denotes that over the faces of the computational cell with the direction of inflow determined in the convection velocity, or the sign of the mass flux. If the time derivative term is removed, the diagonal term is exactly the same as the sum of the off-diagonal terms. The first-

order upwind linearization is used for the changes of the mass fractions on the cell faces. The modified residual is now on the right hand side. In a traditional pressure-correction algorithm, the mass imbalance is assumed to be zero. Since the residuals are calculated simultaneously, the effect of this imbalance must be filtered away (Siikonen, 2011). Now an implicit stage can be written for two-dimensional flow calculations

$$\mathsf{A}_W \Delta x_{i-1} + \mathsf{A}_E \Delta x_{i+1} + \mathsf{A}_S \Delta x_{i-imax} + \mathsf{A}_N \Delta x_{i+imax} + \mathsf{A}_P \Delta x_i = R_i - x_i R_i^m \ , \tag{3.93}$$

where the one-dimensional index notation was used for simplicity. The coefficients of the matrix are

$$\begin{aligned} \mathsf{A}_W &= -\mathsf{max}(\dot{m}_{i-1/2}, 0) \ , \\ \mathsf{A}_E &= -\mathsf{max}(-\dot{m}_{i+1/2}, 0) \ , \\ \mathsf{A}_S &= -\mathsf{max}(\dot{m}_{j-1/2}, 0) \ , \\ \mathsf{A}_N &= -\mathsf{max}(-\dot{m}_{j+1/2}, 0) \quad \text{and} \\ \mathsf{A}_P &= \frac{\mathcal{V}_i \rho_i}{\Delta t} - \mathsf{A}_W - \mathsf{A}_E - \mathsf{A}_S - \mathsf{A}_N \ , \end{aligned} \tag{3.94}$$

where the max functions are again used to account for the flow direction. For three-dimensional situations, additional arrays would be needed for the mass fluxes through the faces $k \pm 1/2$. Formally, these resemble the ones above, of course. The coefficients are exactly the same as the inviscid part of the momentum equations, Eqs. (3.30).

3.2.7 Initial and boundary conditions

In order to obtain solutions to flow cases of interest, one must set the proper boundary conditions. Roughly speaking, it is mainly those that distinguish a flow case from another. Additionally, to begin the computations, the initial solution inside the domain must be known. The initial state is usually set to correspond to undisturbed free surface, that is, the void fraction is initialized according to a given free surface level and the velocity is set to that of the free stream. The boundary conditions that are applied in computational fluid dynamics are usually either of the Dirichlet or von Neumann type (Siikonen,

2008), *i.e.*

$$U = C \qquad \text{and} \tag{3.95}$$

$$\left.\begin{array}{l} \frac{\partial U}{\partial n} = C \\ \frac{\partial U}{\partial s} = C \end{array}\right\} \quad , \tag{3.96}$$

respectively. Above, U is a flow variable, C a known value, n denotes the normal direction of the boundary and s the tangential direction of the boundary. These two types state that either the flow variables are assigned to a known value at the boundaries (the Dirichlet condition), or their derivatives are set to a prescribed value (von Neumann condition).

Unfortunately in practical calculations, the physical domain must be confined such that problems can be simulated in comprehensible computer times. However, the solution is seldom known outside these artificially defined boundaries. Should one require the influences from upstream and downstream locations to diminish, the domain would extend to infinity. Other boundaries are of the physical type, *e.g.* solid surfaces. The boundaries involved in this work are of five different types. These are inflow, outflow, mirror, slip and no-slip boundary conditions, and will be detailed below. Note again that no free surface boundary conditions need addressing here, since there is no free surface.

Inflow boundary condition. The inlet of the computational domain is assumed to be sufficiently far such that undisturbed free stream boundary conditions can be applied. This means that a Dirichlet condition can be imposed for the velocity, or $\vec{V} = (u_{in}, 0, 0)^{\mathsf{T}}$. A von Neumann condition is applied to the longitudinal pressure gradient, such that it is set to zero, or $\partial p/\partial x = 0$ at the inlet boundary. This means that the vertical pressure distribution reduces to the hydrostatic one. The gas volume fraction is set to a prescribed value according to the initial still free surface, or $\alpha(x, y, z) = 0$ if z is in the liquid phase and $\alpha(x, y, z) = 1$, otherwise.

Outflow boundary condition. It is assumed again that the outflow boundary is sufficiently far downstream, *i.e.*, no flow disturbance reaches the boundary. The velocity is linearly extrapolated from the domain. The pressure is set to a pre-described value, which in this case corresponds to the

hydrostatic pressure at the initial still water level. The gas volume fraction is also extrapolated from the computational domain.

Mirror boundary condition. The mirror boundary is imposed on the boundaries in the normal directions of which the flow can be assumed to be symmetric. This requires transformation of the velocity into a local Cartesian coordinate system that has a coordinate direction normal to the boundary face. After the transformation, the velocity components in the computational domain can be mirrored to the ghost cells on the other side of the mirror boundary. The normal gradients of the pressure and the void fraction can again be set to zero.

Slip boundary condition. This corresponds to the inviscid wall boundary condition. The normal component of the velocity is zero, and the tangential velocity component is free to slip along the surface. The gradients of the pressure and the void fraction are set to zero in a direction normal to the slip boundary.

No-slip boundary condition. This corresponds to the viscous wall boundary condition. The velocity is set to zero at the boundary. The gradients of the pressure and the void fraction are again set to zero in the normal direction of the boundary.

The casting of boundary conditions is one of the most important tasks in CFD analyses. In the present computations, the typical boundary conditions have been imposed. In free surface problems in particular, the inflow and outflow conditions are especially sensitive to the flow case, as there may appear non-physical reflections where one would expect a still state. Note also that no attention was given for turbulence quantities. This discussion will be resumed in Chap. 5.

3.3 Numerical method for the convection problem

To further assess the numerical solution to the convection problem, a two-dimensional convection solver is constructed. The discretization is based on structured finite volumes, and the conservative form of the Eq. (2.21) is considered. The convection equation for the void fraction was derived in Sec. 2.2

$$\frac{\partial \alpha}{\partial t} + \frac{\partial \alpha u_j}{\partial x_j} = 0 \ , \tag{3.97}$$

where $u_j \geq 0$ is the velocity component in the j-direction; summation is implied over j. In order to devise a finite-volume scheme for the discretization of Eq. (3.97), the integration of it over an infinitesimal control volume $\mathscr{V}$ is required. Such operation gives

$$\int\limits_{\mathscr{V}} \frac{\partial \alpha}{\partial t} \, d\mathcal{V} + \int\limits_{\partial \mathscr{V}} \alpha \bar{u} \, dS = 0 \ , \tag{3.98}$$

where $\bar{u} = \vec{V} \cdot \vec{n}$ is the convection velocity. This can be put in an approximative form for each discrete control volume $\mathcal{V}_i$, similarly to Eq. (3.6)

$$\mathcal{V}_i \frac{\Delta \alpha_i}{\Delta t} + \sum_{faces} f_j S_j = 0 \ . \tag{3.99}$$

The numerical flux of α through the faces of the computational cell $\mathcal{V}_i$ are $f^*_{i+1/2} = \bar{u}_{i+1/2} \alpha_{i+1/2}$. This can written for two-dimensional problems as

$$\mathcal{V}_{i,j} \frac{\Delta \alpha_{i,j}}{\Delta t} = -F_{i+1/2,j} + F_{i-1/2,j} - F_{i,j+1/2} + F_{i,j-1/2} \ , \tag{3.100}$$

where $F_{i+1/2,j} = f^*(\alpha_{i+1/2,j}) S_{i+1/2,j}$ is the flux on the surface $(i+1/2, j)$ of a computational cell (i, j), for instance. Such is depicted in Fig. 3.7. The difference of the fluxes

$$R_{i,j} = -F_{i+1/2,j} + F_{i-1/2,j} - F_{i,j+1/2} + F_{i,j-1/2} \tag{3.101}$$

represents the explicit residual.

The convection equation of the void fraction, Eq. (3.99), consists of time

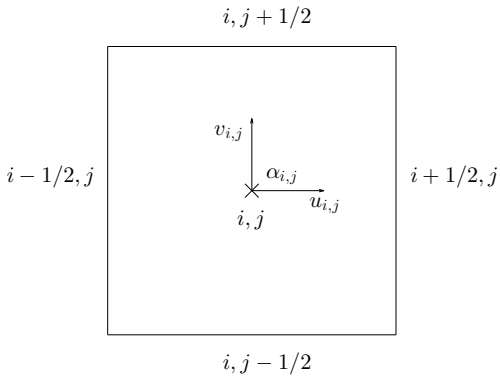


Figure 3.7: Two-dimensional computational cell.

derivatives at the cell centres and numerical fluxes at the faces of the cells. The fluxes are purely convective, and the equations do not have a diffusion nor the source term. discretization of these is done separately. In the following subsections, the discretization processes are detailed and their implementation to the two-dimensional computer program is described.

3.3.1 Temporal discretization

In this section, the implemented time integrations are described. They can be of explicit or implicit nature, the former implying the fact that the time derivative is evaluated by terms from the previous time level, *i.e.*, those values are known. Implicit methods treat certain terms also on the right hand side of the equations as unknowns, that is, they are also evaluated on the new time level. Generally implicit methods are preferred in CFD due to their better coupling and consequently improved stability characteristics over explicit methods. They however require the simultaneous solution of system of equations.

The simple first-order time integration of Eq. (3.99) is the so-called explicit Euler method (cf. Press et al. (1992, p. 704))

$$\mathcal{V}_{i,j} \frac{\Delta \alpha_{i,j}}{\Delta t} = R^n_{i,j} \ , \tag{3.102}$$

where n denotes the time level. The resulting algorithm is first-order accu-

rate in time and has also stability restrictions with the choice of the time increment Δt, or the cell size $\mathcal{V}$. An implicit Euler method can be written as (cf. Press et al. (1992, p. 729))

$$\mathcal{V}_{i,j} \frac{\Delta \alpha_{i,j}}{\Delta t} = R_{i,j}^{n+1} \ , \tag{3.103}$$

where the residual is taken on the new time level. Now the right hand side cannot be computed using the results from the previous time level. The application of an implicit time integration requires linearization of the flux terms that appear on the new time level. This is done as

$$F_{i+1/2,j}^{n+1} \approx F_{i+1/2,j}^{n} + \left. \frac{\partial F^n}{\partial \alpha} \right|_{i+1/2,j} \cdot \Delta \alpha_{i,j}^{n+1} \tag{3.104}$$

with the simple upwind approximation. Note that $\Delta\alpha$ is a vector of length $imax \times jmax$, if $imax$ is the amount of cells in i-direction and $jmax$ the amount of cells in j-direction. The Jacobian is a single element scalar $(\bar{u}S)_{i+1/2,j}$ which in this case is trivial. The method results in bi-diagonal matrix in 1D and prevents the stability issues of its explicit counterpart but is still possesses first-order temporal accuracy. The equations can be written as

$$\mathsf{A} \Delta \alpha = R \ , \tag{3.105}$$

where $\Delta\alpha$ is a column vector of the change and R is a column vector of the explicit residual. The u and v are now positive, so in two dimensions the non-zero elements of the coefficient matrix are

$$\mathsf{A}_{i,i} = \left(\frac{\mathcal{V}_{i,j}}{\Delta t} + (\bar{u}S)_{i+1/2,j} + (\bar{u}S)_{i,j-1/2} \right) \ , \tag{3.106}$$

$$\mathsf{A}_{i,i-1} = -(\bar{u}S)_{i-1/2,j} \quad \text{and} \tag{3.107}$$

$$\mathsf{A}_{i,i-imax} = -(\bar{u}S)_{i,j-1/2} \ , \tag{3.108}$$

where the one-dimensional indexing of the coefficient matrix is adopted for simplicity. The resulting algorithm is no longer bi-diagonal but consist of a diagonal band, one below it and one band a distance of $imax$ to the left of the diagonal due to the upwind linearization. The matrix, though very sparse, of the size of $imax \times jmax$ needs to be in principle inverted for the change $\Delta\alpha$ for every time step. No boundary condition cells need to be considered in this study. The Crank-Nicolson method is obtained if an average is used

(cf. Press et al. (1992, p. 840))

$$\mathcal{V}_{i,j}\frac{\Delta\alpha_{i,j}}{\Delta t} = \frac{1}{2}\left(R^n_{i,j} + R^{n+1}_{i,j}\right) \ , \tag{3.109}$$

which is now second-order accurate in time. The elements of the matrix are simply

$$\mathsf{A}_{i,i} = \left(\frac{\mathcal{V}_{i,j}}{\Delta t} + \frac{1}{2}\left[(\bar{u}S)_{i+1/2,j} + (\bar{u}S)_{i,j-1/2}\right]\right) \ , \tag{3.110}$$

$$\mathsf{A}_{i,i-1} = -\frac{1}{2}(\bar{u}S)_{i-1/2,j} \quad \text{and} \tag{3.111}$$

$$\mathsf{A}_{i,i-imax} = -\frac{1}{2}(\bar{u}S)_{i,j-1/2} \ . \tag{3.112}$$

A higher-order Runge-Kutta time integration method is implemented. This is possible, since Eq. (3.97) reduces to an ODE for each control volume. The implementation described by Tannehill et al. (1997, p. 125) is applied here, that is a fourth-order accurate method (the truncation error is $\mathcal{O}(\Delta t^4)$) the steps of which are

$$\begin{aligned}
\Delta\alpha^{(1)}_{i,j} &= \frac{\Delta t}{2\mathcal{V}_{i,j}} R^n_{i,j} \ , \\
\Delta\alpha^{(2)}_{i,j} &= \frac{\Delta t}{2\mathcal{V}_{i,j}} R^{(1)}_{i,j} \ , \\
\Delta\alpha^{(3)}_{i,j} &= \frac{\Delta t}{\mathcal{V}_{i,j}} R^{(2)}_{i,j} \quad \text{and} \\
\Delta\alpha_{i,j} &= \frac{\Delta t}{6\mathcal{V}_{i,j}}\left(R^n_{i,j} + 2R^{(1)}_{i,j} + 2R^{(2)}_{i,j} + R^{(3)}_{i,j}\right) \ .
\end{aligned} \tag{3.113}$$

The notations $\Delta\alpha^{(m)}$ and $R^{(m)}$ denote the temporary change and residual from the values of m:th step, *i.e.* $R^{(m)} = R\left(\alpha^{(m)}\right)$. Four loops within a time step are required. Additional logical statements are needed for the updating of the change $\Delta\alpha$.

After solving for the change $\Delta\alpha$ by whichever method described above, the new time level solution is calculated

$$\alpha^{n+1} = \alpha^n + \Delta\alpha \ , \tag{3.114}$$

where α^n denotes the solution from the previous time level.

3.3.2 Spatial discretization

This section focuses on the discretization of the flux term at the cell faces. The void fraction is convected only through the faces with the convection velocity $\bar{u} = \vec{V} \cdot \vec{n}$. In this study, the velocity field is predescribed. The computation of the flux on a cell face from the nodal values is a fundamental problem in computational fluid dynamics. In principle, this averaging process can be done in two ways. First is direct extrapolation (or interpolation) of the dependent variable to the cell face. Another way is to compute the fluxes at the nodal points as well, after which they are somehow extrapolated to the cell faces. The first approach is used here. Special procedures are needed for the discretization of the convection of α as was described in Sec. 2.2. Here, the TVD (total variation diminishing), and NVF (normalized variable formulation) using the CBC (convective boundedness criterion) discretization strategies are used.

The analytical flux of the void fraction is

$$\mathscr{F} = \alpha \bar{u} \ . \tag{3.115}$$

The discrete representation of this on the surface of the computational cell $i + 1/2$ is the so-called numerical flux $f^*_{i+1/2}$. It is a function of the values of α at both sides of the face, or

$$f^*_{i+1/2} = f^*(\alpha_i, \alpha_{i+1}) \ . \tag{3.116}$$

The consistency of the numerical flux with the analytical one is of great importance (Davis, 1994; Siikonen, 2008; Tannehill et al., 1997). This is enforced through the condition

$$f^*(\alpha, \alpha) = \mathscr{F}(\alpha) \ . \tag{3.117}$$

A higher-order numerical flux is functionally

$$f^*_{i+1/2} = f^*(\alpha_l, \alpha_r) \ , \tag{3.118}$$

where the sub-indexes l and r denote the left and right sides of the faces $i+1/2$. A second-order approximation promotes the fact that now the values of α may vary linearly inside a cell; the first-order flux function assumes that

Table 3.3: Some MUSCL differencing schemes.

Extrapolation	κ
2nd-order upwind	-1
Fromm's method	0
3rd-order upwind biased	1/3
QUICK	1/2
Central difference	1

the states are constant within each cell. If one is to employ the Roe scheme for the numerical flux function, it turns out that

$$\begin{aligned} \bar{A} &= \frac{\mathscr{F}(\alpha_r) - \mathscr{F}(\alpha_l)}{\alpha_r - \alpha_l} \\ &= \bar{u} \ , \end{aligned} \tag{3.119}$$

where $\bar{A}$ is the mean value matrix (cf. Davis (1994); Tannehill et al. (1997)). It follows from the approximate Riemann solver of Roe, and is of use also in the hyperbolic Euler system, for instance. Now the higher-order numerical flux function reduces simply to

$$f^*_{i+1/2} = \bar{u}\alpha^l_{i+1/2} \ , \tag{3.120}$$

where $\alpha^l_{i+1/2}$ is the value of the void fraction on the left side of the cell face $i + 1/2$. The explicit indication of the left side is hereafter dropped. Now, this form ensures that the correct jump is recovered when a discontinuity is encountered and the correct solution in regions where α is smooth (Tannehill et al., 1997).

The flux on the cell face is obtained from the dependent variable, but they are known only on the cell centres. Traditionally the cell face values are calculated with the aid of the MUSCL formula

$$\alpha_{i+1/2} = \alpha_i + \frac{1}{4}\big((1-\kappa)(\alpha_i - \alpha_{i-1}) + (1+\kappa)(\alpha_{i+1} - \alpha_i)\big) \ , \tag{3.121}$$

where the variable is extrapolated or interpolated from the values of the neighbouring cell centres, depending on the value of the parameter κ. Several well-known schemes can be obtained with different values of the parameter κ, and these are listed in Tab. 3.3. All methods for which $-1 \leq \kappa \leq 1$ approximate $\alpha_{i+1/2}$ at least to second-order accuracy (Siikonen, 2008). As was

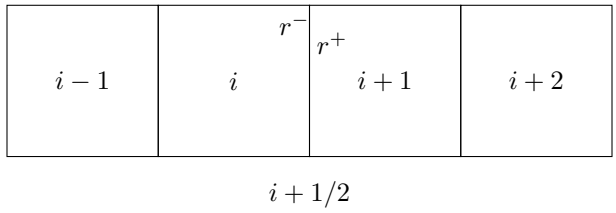


Figure 3.8: Definitions of the different sided slope ratios, Eq. (3.125).

demonstrated previously, any higher-order spatial discretization inevitably includes oscillations in the approximation of discontinuous profiles. Such is the distribution of the void fraction α. These are unphysical and usually completely spoil the calculations by introducing new extrema into the numerical solution. Again, first-order methods, whilst monotone, smear the solution over multiple cells. In this study, two commonly known strategies are implemented to tackle these, and they are the TVD and NVF discretization methods.

Total variation diminishing

It has been shown that *total variation* of a physically possible solution for a scalar conservation law does not increase in time (Harten, 1983; Sweby, 1984). The total variation is defined by

$$TV = \int \left| \frac{\partial \alpha}{\partial x} \right| \, dx \ , \tag{3.122}$$

which is for discrete cases

$$TV(\alpha) = \sum_j |\alpha_{j+1} - \alpha_j| \ . \tag{3.123}$$

A numerical method is said to have the property of total variation diminishing, or TVD, if it satisfies the condition

$$TV(\alpha^{n+1}) \leq TV(\alpha^n) \ . \tag{3.124}$$

The crucial fact is that such method is *monotonicity* preserving. When a solution is known to contain very steep gradients or discontinuities, the order

of discretization is lowered in those regions in order to prevent the oscillations (first-order methods are monotone, cf. Fig. 2.2(a)), and higher-order accuracy is retained in smooth regions. This type of switching introduces a non-linear dependency of the discretization scheme on the solution itself. This is accomplished by the use of limiters. That is, the discretization is based on local ratios

$$r^+_{i+1/2} = \frac{\alpha_{i+2} - \alpha_{i+1}}{\alpha_{i+1} - \alpha_i} \quad \text{and} \quad r^-_{i+1/2} = \frac{\alpha_i - \alpha_{i-1}}{\alpha_{i+1} - \alpha_i} \ , \tag{3.125}$$

which are the different sided gradients near the face $i+1/2$, see Fig. 3.8. The consecutive differences are of different sign in presence of discontinuities. The construction of a TVD scheme relies on choosing a function $\phi(r)$ such that it adds the limited anti-diffusive flux that originates from, *e.g.*, the second-order accurate Lax-Wendroff scheme to the monotone first-order scheme, and does this in such a way that the anti-diffusion does not violate the convergence properties[1] of the scheme (Sweby, 1984). The function ϕ is called a limiter function. It is subject to the TVD constraints, which have been shown to be

$$0 \leq \left(\frac{\phi(r)}{r}, \phi(r) \right) \leq 2 \ . \tag{3.126}$$

Second-order accuracy is further assured by demanding that the limiter function lays within the range set by so-called Warming-Beam and Lax-Wendroff schemes, for which $\phi = r$ and $\phi = 1$, respectively; it can be shown that any TVD scheme relying on points in Fig. 3.8 can be constructed as a linear combination of these two. Details are given by Sweby (1984), Tannehill et al. (1997) and Siikonen (2008), for instance. The values of $\phi > 1$ produce artificial compression in the extrapolation. The Sweby diagram displaying the TVD area, Eq. (3.126), together with the second-order TVD region, are plotted in Fig. 3.9. Some limiter functions ϕ that are of use here will be given later. The MUSCL formula (3.121) can be written with limited upwind and central parts as

$$\alpha_{i+1/2} = \alpha_i + \frac{1}{4}\left((1-\kappa)\phi\left(1/r^-_{i+1/2}\right)(\alpha_i - \alpha_{i-1}) + (1+\kappa)\phi\left(r^-_{i+1/2}\right)(\alpha_{i+1} - \alpha_i) \right) . \tag{3.127}$$

The chosen extrapolation scheme (value for κ) does not affect the limited value for α in regions where the limiter is active (Siikonen, 2008). The

[1] The convergence of this non-linear coupling can however be strictly proven for the explicit methods in one space dimension (Jasak, 1996).

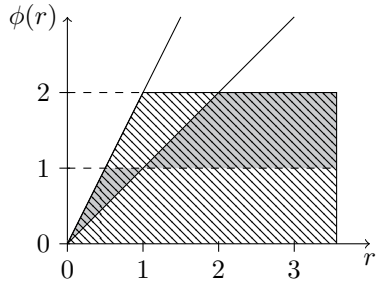


Figure 3.9: The Sweby diagram with the TVD area shaded by the diagonal lines. The second-order TVD region is coloured with grey.

implementation is convenient in this way, and considering overall accuracy, the third-order scheme is used.

Normalized variable formulation

Now introducing the *normalized variable* of Leonard (1988, 1991), that is

$$\tilde{\alpha} = \frac{\alpha - \alpha_{i-1}}{\alpha_{i+1} - \alpha_{i-1}} \tag{3.128}$$

or a definition of a dimensionless variable that depends on two differences, one including its own nodal position and an upwind node and the other comprising the farthest nodes in the molecule. One can then define so-called normalized variable diagram (NVD). The one-dimensional normalizations of $\alpha_i, \alpha_{i+1}, \alpha_{i-1}$ and $\alpha_{i+1/2}$ are plotted in Fig. 3.10.

With this new definition, it can be seen that

$$\tilde{\alpha}_{i+1/2} = \mathfrak{f}(\tilde{\alpha}_i) \tag{3.129}$$

only (Jasak et al., 1999; Leonard, 1991). In fact, the MUSCL scheme, Eq. (3.121), can now be written in terms of the normalized $\tilde{\alpha}_i$ as

$$\tilde{\alpha}_{i+1/2} = \tilde{\alpha}_i + \frac{1}{4}\big((1-\kappa)\tilde{\alpha}_i + (1+\kappa)(1-\tilde{\alpha}_i)\big) \ , \tag{3.130}$$

since the far upwind and downwind nodes are known in their normalized representation. By inspecting Fig. 3.10, the normalized variable satisfies the

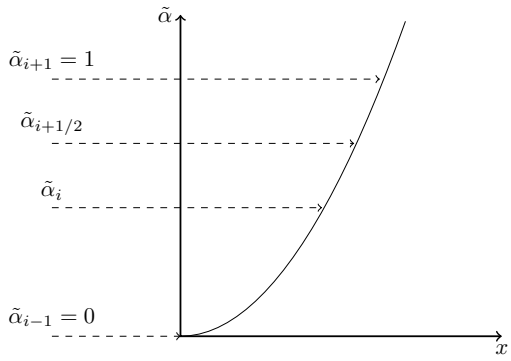


Figure 3.10: Normalized variable $\tilde{\alpha}$.

monotonicity condition, that is, it does not produce new extrema in the solution if certain criteria are met. Using the normalized variable formulation, the so-called convective boundedness criteria (CBC), usually accredited to Gaskell and Lau (1988), can be written as

$$\begin{aligned} \tilde{\alpha}_{i+1/2} = \tilde{\alpha}_i \quad &\text{for} \quad \begin{cases} \tilde{\alpha}_i < 0 \\ \tilde{\alpha}_i > 1 \end{cases} \quad \text{and} \\ \tilde{\alpha}_i \leq \tilde{\alpha}_{i+1/2} \leq 1 \quad &\text{for} \quad 0 \leq \tilde{\alpha}_i \leq 1 \quad , \end{aligned} \tag{3.131}$$

which apply to steady-state problems. The lower limit is the first-order upwind, the most diffusive, and the upper limit the first-order downwind, the most compressive, extrapolation. The latter condition changes in transient calculations to

$$\tilde{\alpha}_i \leq \tilde{\alpha}_{i+1/2} \leq \mathsf{min}\left(1, \frac{\tilde{\alpha}_i}{\mathsf{C}}\right) \quad \text{for} \quad 0 \leq \tilde{\alpha}_i \leq 1 \ , \tag{3.132}$$

which is given by Leonard (1991) and Ubbink (1997). The conditions from Eqs. (3.131 – 3.132) are shown in Figs. 3.11 and 3.12. The former is plotted together with the various MUSCL schemes given in Tab. 3.3. All second-order schemes go through the point $(1/2, 3/4)$. The expressions (3.131 – 3.132) are the necessary conditions for monotonicity on $\tilde{\alpha}_{i+1/2}$, as noted by Leonard (1991), Ubbink (1997), Ubbink and Issa (1999), Jasak (1996), Jasak and Weller (1995), Jasak et al. (1999) and Darwish and Moukalled (2006), for instance. The lower bound for the CBC is the first-order upwind extrapolation, or $\tilde{\alpha}_{i+1/2} = \tilde{\alpha}_i$, and the upper bound the first-order downwind

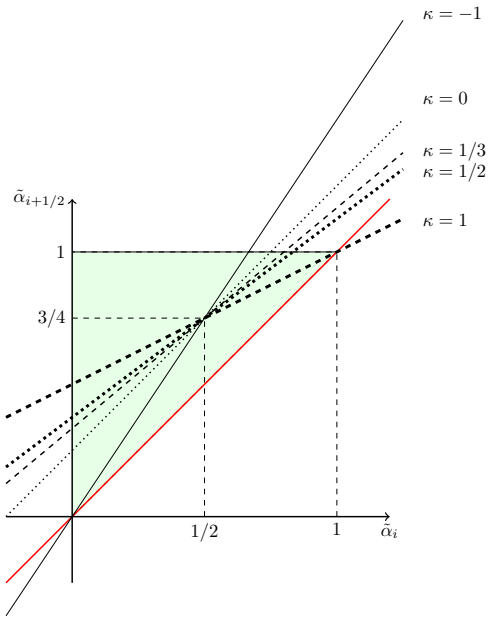


Figure 3.11: Normalized variable diagram (NVD) and various MUSCL schemes. The shaded area denotes bounded region, and the red line the first-order upwind extrapolation.

extrapolation, $\tilde{\alpha}_{i+1/2} = 1$. It should be noted that the explicit CBC result in the exact solution of Eq. (2.21) with $\mathsf{C} = 1$.

From the above it is evident that the gradient ratio $r^-_{i+1/2}$ used in construction of a scheme fulfilling the TVD criterion is analogous to the normalized variable $\tilde{\alpha}_i$, using which the CBC were set. In addition since the MUSCL type extrapolation can be recast in terms of the normalized variable, the equivalent expression for limited MUSCL schemes is

$$\tilde{\alpha}_{i+1/2} = \tilde{\alpha}_i + \frac{1}{4}\left((1-\kappa)\phi\left(1/r^-_{i+1/2}\right)\tilde{\alpha}_i + (1+\kappa)\phi\left(r^-_{i+1/2}\right)(1-\tilde{\alpha}_i)\right) . \quad (3.133)$$

The TVD conditions can also be cast with the normalized variables, and these are

$$\begin{aligned} \tilde{\alpha}_{i+1/2} = \tilde{\alpha}_i \quad &\text{for} \quad \begin{cases} \tilde{\alpha}_i < 0 \\ \tilde{\alpha}_i > 1 \end{cases} \quad \text{and} \\ \tilde{\alpha}_i \le \tilde{\alpha}_{i+1/2} \le \mathsf{min}(2\tilde{\alpha}_i, 1) \quad &\text{for} \quad 0 \le \tilde{\alpha}_i \le 1 \quad , \end{aligned} \quad (3.134)$$

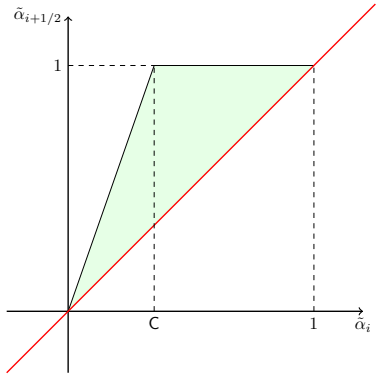


Figure 3.12: Normalized variable diagram (NVD) for transient calculations. The shaded area denotes bounded region for transient flow calculations (Leonard, 1991; Ubbink, 1997) and C the Courant number on the cell face $(i + 1/2)$. The red line shows the first-order upwind extrapolation.

which are given by Leonard (1991). These are plotted in Fig. 3.13. One should note that the TVD conditions are more stringent in terms of the allowable cell face values than the CBC. The NV approach in itself does not guarantee the convergence of the differencing scheme, even on one-dimensional problems (Jasak, 1996).

The blending[2] strategy

Moreover, it is common practice to base some discretization schemes to the angle γ between the interface and the cell face normals (Darwish and Moukalled, 2006; Lafaurie et al., 1994; Ubbink, 1997; Wackers, 2007). That angle is defined in Fig. 3.14. This practice stems from the donor-acceptor strategy introduced by Hirt and Nichols (1981), now switching to a proper discretization for the cell face value of the void fraction such that it follows the composition of the cells surrounding it more accurately. In other words, the blending option aids in the choice of a more compressive scheme when this is necessary, and reverts to higher-order discretization when their diffusion is not a problem, since overly compressive schemes, such as the pure first-order downwind scheme, tend to wrinkle the interface (Lafaurie et al.,

[2]This should not be confused with the blending function f_x mentioned in Sec. 3.2.2. There, it denoted the need of blending between different schemes depending upon $\tilde{\alpha}$ as in Tabs. 3.1 and 3.2, without the angle γ, arising solely from the usage of arbitrary grids. In this work, arbitrary grids are excluded.

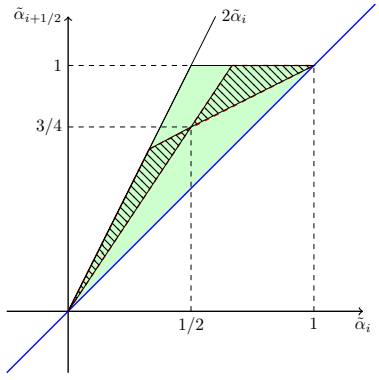


Figure 3.13: The TVD criteria plotted using the normalized variable. The TVD region is shaded with green, and the second-order TVD region is shaded by the diagonal lines. The blue line shows the first-order upwind extrapolation.

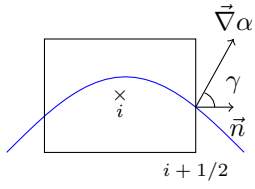


Figure 3.14: The angle between the cell face $i + 1/2$ and the interface normal $\vec{\nabla}\alpha$.

1994; Leonard, 1991; Ubbink, 1997).

This can be illustrated with Fig. 3.15. Following Darwish and Moukalled (2006), the need for a compressive discretization arises in situations where the donor cell has started to fill from the upwind side and the interface is parallel to the cell face normal, which is shown in Fig. 3.15(a). This is to say that only the fluid that is present in the downstream cell should convect through the cell face, denoted as f in the figure. If the interface is perpendicular to the cell face as in Fig. 3.15(b), the situation is the opposite, and a high resolution scheme should be used. The case of Fig. 3.15(c) would allow for either of the schemes, whereas blending of the compressive and high-resolution schemes is needed in configurations where $0 < \gamma < 90$ as in Fig. 3.15(d). The blending is based on expressions of the type

$$\tilde{\alpha}_{i+1/2} = f(\gamma) \cdot \tilde{\alpha}^{s_1}_{i+1/2} + (1 - f(\gamma)) \cdot \tilde{\alpha}^{s_2}_{i+1/2} \ , \tag{3.135}$$

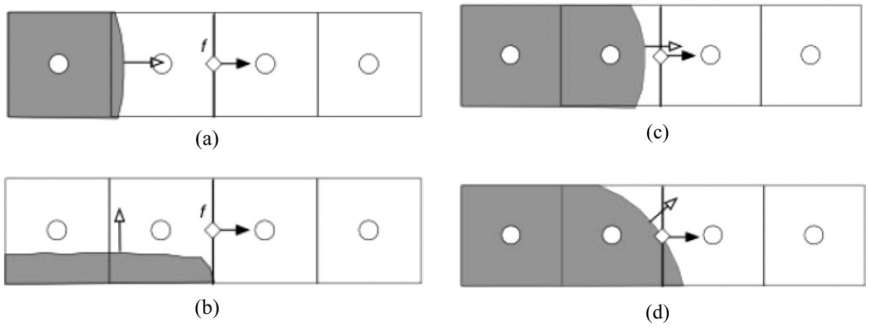


Figure 3.15: Different flow configurations when either a compressive (a), a high-resolution (b), either of these two (c), or blending of the schemes (d) based on the angle of the interface and the cell face is needed (Darwish and Moukalled, 2006).

where $f(\gamma)$ is the blending function, and s_i denote different discretization schemes taken for respective $\tilde{\alpha}$. This way, families of high-resolution and compressive schemes can be easily devised for interface capturing algorithms.

In order to obtain the angle γ, one needs the gradient of the void fraction since this is the normal of the interface. The angle γ follows from the expression

$$\cos(\gamma_{i+1/2}) = \frac{\vec{\nabla}\alpha_{i+1/2} \cdot \vec{n}_{i+1/2}}{|\vec{\nabla}\alpha_{i+1/2}| \; |\vec{n}_{i+1/2}|} \; . \tag{3.136}$$

It must be noted here that the computation of this angle is not a well-defined task since the distribution of the void fraction is supposed to be a step function. It could be watered down in a steady-state problem (cf. Chap. 5); now it is computed using the particular simplicity of the uniform grid.

Implemented discretization schemes

Finally, several different discretization schemes for the convection of the void fraction were implemented. The choice of these was based on a revision of the recent development of a few promising schemes, as well as studying the schemes applied in Sec. 3.2. Hence, two TVD schemes and six schemes based on the NVF were implemented. The former are plotted in the Sweby diagram and the latter in the normalized variable diagram; the relationship of these two visualzations is shown in Fig. 3.13.

The implemented TVD schemes are the limiter of van Albada (Siikonen, 2008) and the SUPERBEE limiter of Roe (Leonard, 1991; Siikonen, 2008). These are plotted in Fig. 3.16, and they have the expressions

$$\phi_{va} = \frac{r + r^2}{1 + r^2} \tag{3.137}$$

and

$$\phi_{sb} = \mathsf{max}\big(0, \mathsf{min}(2r, 1), \mathsf{min}(r, 2)\big) \ , \tag{3.138}$$

respectively.

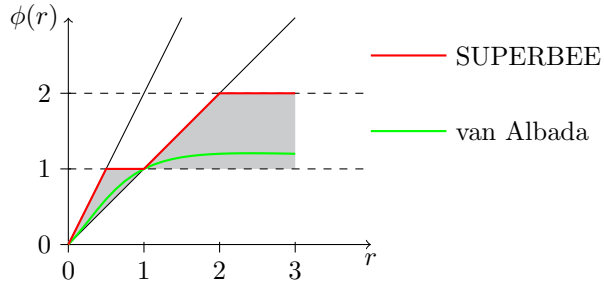


Figure 3.16: Plots of the van Albada and SUPERBEE limiters in the Sweby diagram.

The following schemes based the NVF were implemented. Second- and Third-Order Interpolation for Convection (STOIC) of Darwish (1993), Compressive Interface Capturing Scheme for Arbitrary Meshes (CICSAM) of Ubbink (1997), Switching Technique for Advection and Capturing of Surfaces (STACS) of Darwish and Moukalled (2006), Fast Interface Capturing Scheme (FICS) of Schmonde et al. (2007), Inter-Gamma Differencing Scheme (IGDS) of Jasak and Weller (1995) and Blended Interface Capturing Scheme (BICS) of Wackers et al. (2011).

The schemes that are based on the NVF are given below with their NVD. Only the schemes that explicitly involve the local Courant number are shown with such dependency in the respective NVD.

The STOIC is an un-blended scheme given by

$$\tilde{\alpha}_{i+1/2}^{stoic} = \begin{cases} \tilde{\alpha}_i & \text{for } \tilde{\alpha}_i \leq 0 \quad , \\ \frac{1}{2} + \frac{1}{2} \cdot \tilde{\alpha}_i & \text{for } 0 < \tilde{\alpha}_i \leq \frac{1}{2} \quad , \\ \frac{3}{8} + \frac{3}{4} \cdot \tilde{\alpha}_i & \text{for } \frac{1}{2} < \tilde{\alpha}_i \leq \frac{5}{6} \quad , \\ 1 & \text{for } \frac{5}{6} < \tilde{\alpha}_i \leq 1 \quad \text{and} \\ \tilde{\alpha}_i & \text{for } 1 < \tilde{\alpha}_i \quad , \end{cases} \tag{3.139}$$

where the second scheme corresponds to central differencing and the third to the QUICK scheme. It is noted that this implementation differs by the absence of an "ad hoc linear function in the [0-0.2] segment" as was used in the original STOIC by Darwish (1993). The use of the linear function is not necessary here, due to the Courant numbers of the problem given in the next chapter. More recently application by Darwish and Moukalled (2006) omits this altogether due to the implicit transient discretization; also here the lowest $\mathsf{C} = 0.2$. The STOIC is composed of the central difference, QUICK and the first-order downwind schemes, in addition to the first-order upwind when $\tilde{\alpha}_i \leq 0$ or $\tilde{\alpha}_i > 1$. The NVD of STOIC is shown in Fig. 3.17.

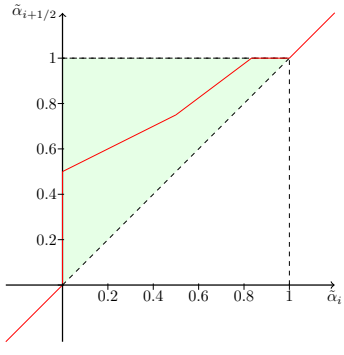
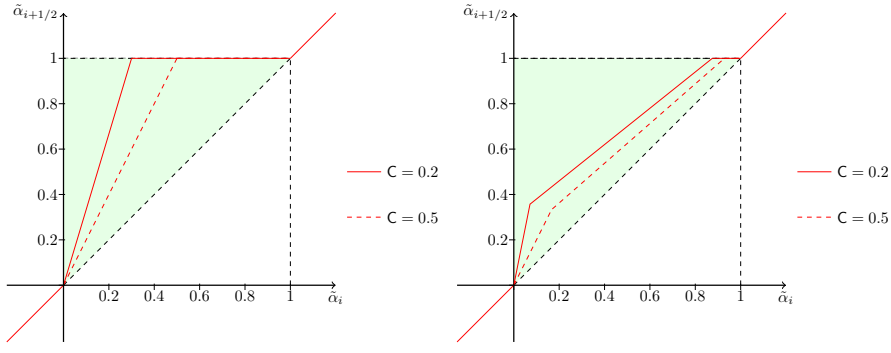


Figure 3.17: STOIC in the normalized variable diagram.

The CICSAM scheme is a blended one, comprising of the upper limit of the CBC

$$\tilde{\alpha}_{i+1/2}^{cbc} = \begin{cases} \tilde{\alpha}_i & \text{for } \tilde{\alpha}_i \leq 0 \quad , \\ \min\left(1, \frac{\tilde{\alpha}_i}{\mathsf{C}}\right) & \text{for } 0 < \tilde{\alpha}_i \leq 1 \quad \text{and} \\ \tilde{\alpha}_i & \text{for } 1 < \tilde{\alpha}_i \quad . \end{cases} \tag{3.140}$$

The transient CBC is actually the HYPER-C scheme (Leonard, 1991), which is the most compressive scheme, *i.e.* the limited first-order downwind. The second scheme is a mixture of the so-called ULTIMATE-QUICKEST (UQ) and the previous one



(a) HYPER-C (b) ULTIMATE-QUICKEST

Figure 3.18: Schemes used in the CICSAM discretization in the normalized variable diagram.

$$\tilde{\alpha}_{i+1/2}^{uq} = \begin{cases} \tilde{\alpha}_i & \text{for } \tilde{\alpha}_i \leq 0 \quad , \\ \min\left(\frac{8\mathsf{C}\cdot\tilde{\alpha}_i+(1-\mathsf{C})(6\cdot\tilde{\alpha}_i+3)}{8}, \tilde{\alpha}_{i+1/2}^{cbc}\right) & \text{for } 0 < \tilde{\alpha}_i \leq 1 \ , \\ \tilde{\alpha}_i & \text{for } 1 < \tilde{\alpha}_i \quad . \end{cases} \tag{3.141}$$

The blending function is

$$f_{cicsam} = \min\left(k_c \frac{\cos(2\gamma)+1}{2}, 1\right) \ , \tag{3.142}$$

where k_c is a constant that was introduced to control the dominance of different schemes; $k_c = 1$ increases the influence of the most compressive HYPER-C and the choice $k_c = 0$ reduces the scheme to the UQ. Ubbink (1997) recommends $k_c = 1$ which was adopted here. The CICSAM scheme is then

$$\tilde{\alpha}_{i+1/2^{cicsam}} = f_{cicsam} \cdot \tilde{\alpha}_{i+1/2}^{cbc} + (1 - f_{cicsam}) \cdot \tilde{\alpha}_{i+1/2}^{uq} \ . \tag{3.143}$$

The NVD of CICSAM is shown in Fig. 3.18. The original CICSAM features a complex predictor-corrector sequence (Ubbink, 1997; Ubbink and Issa, 1999); here, the path of Darwish and Moukalled (2006) is followed in the

implementation.

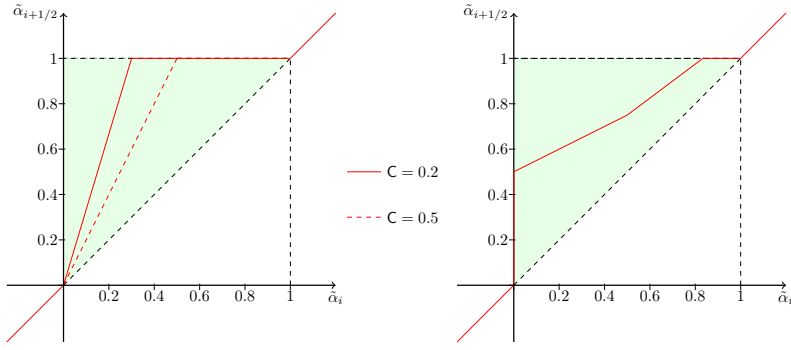
The STACS is again a blended scheme, and comprises of the HYPER-C and STOIC schemes, or

$$\tilde{\alpha}^{stacs}_{i+1/2} = f_{stacs} \cdot \tilde{\alpha}^{hyper-c}_{i+1/2} + (1 - f_{stacs}) \cdot \tilde{\alpha}^{stoic}_{i+1/2} \tag{3.144}$$

with the blending function of

$$f_{stacs} = \mathsf{cos}^4(\gamma) \ . \tag{3.145}$$

The NVD of STACS is shown in Fig. 3.19.



(a) HYPER-C. (b) STOIC.

Figure 3.19: Schemes used in the STACS discretization in the normalized variable diagram.

The FICS consists of modes where the interface is parallel (p) or normal (n) to the cell face, or

$$\tilde{\alpha}^{p}_{i+1/2} = \begin{cases} \tilde{\alpha}_i & \text{for } \tilde{\alpha}_i \leq 0 \quad , \\ \mathsf{min}\,(10 \cdot \tilde{\alpha}_i, 1) & \text{for } 0 < \tilde{\alpha}_i < 1 \quad \text{and} \\ \tilde{\alpha} & \text{for } 1 \leq \tilde{\alpha}_i \quad , \quad \text{and} \end{cases} \tag{3.146}$$

$$\tilde{\alpha}^{n}_{i+1/2} = 0.45 + 0.55 \cdot \tilde{\alpha}_{i+1/2} \ , \tag{3.147}$$

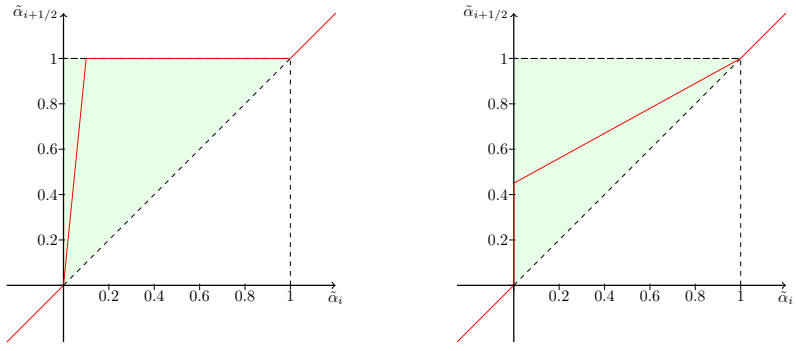
respectively. The parallel mode reduces quickly to the first-order downwind, and the normal resembles the central difference. The scheme is then

$$\tilde{\alpha}^{fics}_{i+1/2} = f_{fics} \cdot \tilde{\alpha}^{p}_{i+1/2} + (1 - f_{fics}) \cdot \tilde{\alpha}^{n}_{i+1/2} \tag{3.148}$$

and its blending function

$$f_{fics} = \cos^2(\gamma) \ . \tag{3.149}$$

The NVD of FICS is shown in Fig. 3.20.



(a) Parallel mode. (b) Normal mode.

Figure 3.20: Schemes used in the FICS discretization in the normalized variable diagram.

The IGDS is an unblended scheme and given by

$$\tilde{\alpha}_{i+1/2}^{igds} = \begin{cases} \tilde{\alpha}_i & \text{for } \tilde{\alpha}_i \leq 0 \quad , \\ -2 \cdot \tilde{\alpha}_i^2 + 3 \cdot \tilde{\alpha}_i & \text{for } 0 < \tilde{\alpha}_i < \frac{1}{2} \quad , \\ 1 & \text{for } \frac{1}{2} < \tilde{\alpha}_i < 1 \quad \text{and} \\ \tilde{\alpha}_i & \text{for } 1 \leq \tilde{\alpha}_i \quad , \end{cases} \tag{3.150}$$

introducing a quadratic profile in the second term, and the first-order downwind for the third. The NVD of IGDS is shown in Fig. 3.21.

The first half of the BICS is given by

$$\tilde{\alpha}_{i+1/2}^{bics^*} = \begin{cases} \tilde{\alpha}_i & \text{for } \tilde{\alpha}_i < 0 \quad , \\ \frac{1-p}{\beta^2}\tilde{\alpha}_i^2 + \left(p + \frac{2(1-p)}{\beta}\right) \cdot \tilde{\alpha}_i & \text{for } 0 \leq \tilde{\alpha}_i \leq \beta \quad , \\ p \cdot \tilde{\alpha}_i + 1 - p & \text{for } \beta < \tilde{\alpha}_i \leq 1 \quad \text{and} \\ \tilde{\alpha}_i & \text{for } 1 < \tilde{\alpha}_i \quad , \end{cases} \tag{3.151}$$

where the new terms are

$$p(\mathsf{C}) = \alpha_p(\mathsf{C}) \cdot p_{igds} + (1 - \alpha_p(\mathsf{C})) \cdot p_{gds} \tag{3.152}$$

and the coefficient

$$\alpha_p = \begin{cases} 1 & \text{for } \mathsf{C} \leq 0.3 \quad \text{and} \\ \frac{\mathsf{C}-0.3}{\exp(\mathsf{C}-0.3)-1} & \text{for } \mathsf{C} > 0.3 \quad , \end{cases} \tag{3.153}$$

in addition to $p_{gds} = 1/2$ and $p_{igds} = 0$. β has a linear dependency on p

$$\beta(p) = a_0 + a_1 \cdot p \ , \tag{3.154}$$

where $a_1 = (\beta_{gds} - \beta_{igds})/(p_{gds} - p_{igds})$, $a_0 = \beta_{igds} - a_1 \cdot p_{gds}$ together with $\beta_{gds} = 0.1$ and $\beta_{igds} = 0.5$. The scheme is blended, this time with the GDS of Jasak (1996)

$$\tilde{\alpha}^{bics}_{i+1/2} = f_{bics} \cdot \tilde{\alpha}^{bics^*}_{i+1/2} + (1 - f_{bics}) \cdot \tilde{\alpha}^{gds}_{i+1/2} \ , \tag{3.155}$$

and the blending function for this scheme is

$$f_{bics} = \sqrt{|\cos(\gamma)|} \ . \tag{3.156}$$

The GDS is given by

$$\tilde{\alpha}^{gds}_{i+1/2} \quad = \quad \begin{cases} \tilde{\alpha}_i & \text{for } \tilde{\alpha}_i \leq 0 \quad , \\ -\frac{\tilde{\alpha}_i^2}{2\beta} + \left(1 + \frac{1}{2\beta}\right) \cdot \tilde{\alpha}_i & \text{for } 0 < \tilde{\alpha}_i < \beta \quad , \\ \frac{1}{2} + \frac{1}{2}\tilde{\alpha}_i & \text{for } \beta \leq \tilde{\alpha}_i < 1 \quad \text{and} \\ \tilde{\alpha}_i & \text{for } 1 \leq \tilde{\alpha}_i \quad , \end{cases} \tag{3.157}$$

introducing quadratic profile in the second term, and the central difference in the third. The coefficient β is a constant for the scheme for which Jasak (1996) recommends $1/10 \leq \beta \leq 1/2$, here, the value of $\beta = 1/2$ is chosen. The NVD of BICS is shown in Fig. 3.22. The first half resembles the IGDS, but introduces more diffusive extrapolations when the Courant number is increased. The GDS is also close to the lower limit of the second-order TVD criterion.

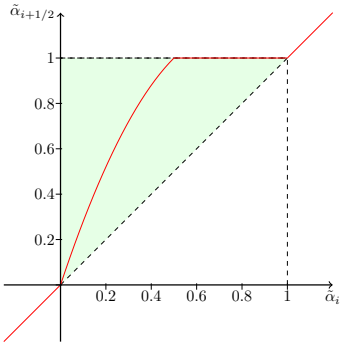
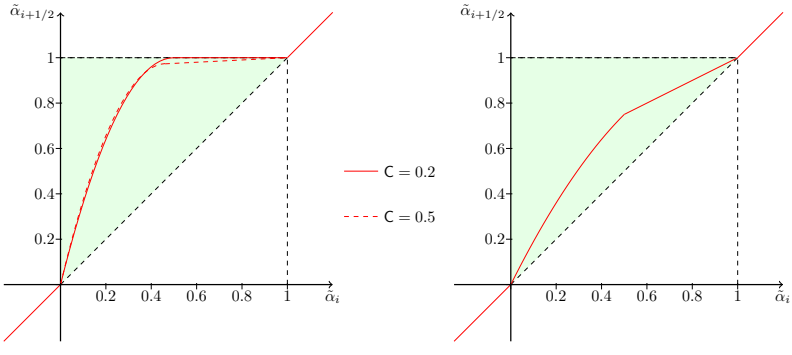


Figure 3.21: IGDS in the normalized variable diagram.



(a) BICS*. (b) GDS.

Figure 3.22: Schemes used in the BICS discretization in the normalized variable diagram.

Chapter 4

Results

4.1 Convection of a step profile in oblique velocity field

This section presents the flow of a step profile in an oblique velocity field. The computations are done with the developed two-dimensional convection equation solver. The solver was described in Sec. 3.3. The purpose of this study is to assess and obtain acquaintance to few recent schemes and thus provide experiences and information upon which such can be chosen for the Navier-Stokes solver. Additionally, the performance of the discretization schemes used in the computation of flow over the submerged ground elevation in Sec. 4.2 is studied in this section.

The cases chosen for the study in this section differ slightly from some 2D test cases used in the literature. For example, Ubbink and Issa (1999), Darwish and Moukalled (2006), Moukalled and Darwish (2012) and Rudman (1997) take hollow squares and hollow and slanted circles, among others, which are convected in an oblique velocity field. Ubbink (1997), Yu et al. (2001), Gao et al. (2012), Wei et al. (2003), Hoekstra et al. (2007), Lafaurie et al. (1994) study similar types of problems as was chosen here, among many others, but focus on these rather qualitatively. The same problem, although with differing parameters, was considered by Ubbink (1997), Yu et al. (2001), Hoekstra et al. (2007) and Lafaurie et al. (1994). Here, the

intention is to study the different well-known discretization schemes for the convection equation of α, and to examine their behaviour for the problem at hand. Rather than comparing to the reference cases, focus is put on the different spatial and temporal schemes that have been implemented in the code. The motivation in the previous studies have mainly originated from the behest of proposing a new family of discretization schemes.

4.1.1 Problem description and computational conditions

The problem has relatively simple characteristics. The velocity field is predescribed as $\vec{V} = (u_0, v_0)$. The velocity vector makes an angle of θ with the x-axis. The scalar profile is a step as depicted in Fig. 4.1. Two shapes, a square located between $x = [0.05, 0.25]$ m and $y = [0.05, 0.25]$ m and a circle centred at $(0.15, 0.15)$ m with a radius of 0.1 are considered. These are shown in Figs. 4.2 and 4.3, respectively. The problem is solved using Courant numbers of $\mathsf{C}_{max} = [0.2, 0.5]$, with the Courant number defined as

$$\mathsf{C}_i = \frac{u_i \Delta t}{\Delta x_i} \tag{4.1}$$

where the index i denotes the coordinate direction. Three grid resolutions

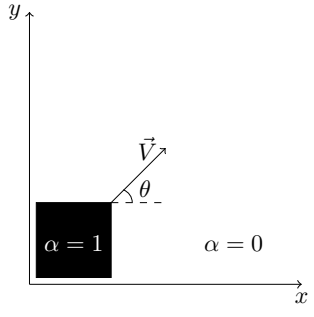


Figure 4.1: Schematic figure of the two-dimensional convection problem.

are used with 50×50, 100×100 and 150×150 cells in total. Furthermore, two different angles $\theta = [15°, 45°]$ are considered. The former is approximately the same as was the observed maximum in the flow over the submerged ground elevation; that case is described in the next section. If the velocity vector forms a 45° angle with the grid lines, the numerical diffusion is at its

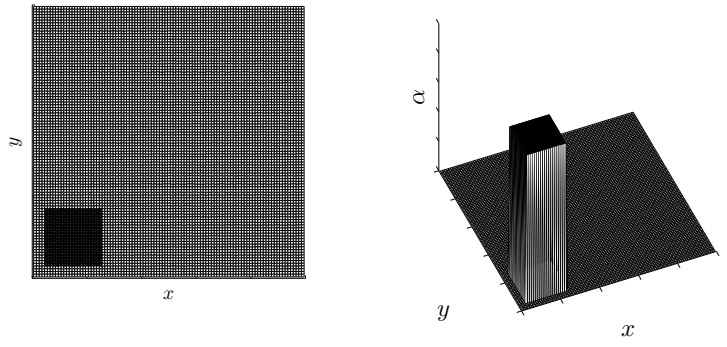


Figure 4.2: Square-shaped step profile.

maximum with standard schemes, and poses a very challenging problem for the numerical solution of the convection equation.

The profile is convected a distance of 0.5 m in x and y-directions for the cases where $\theta = 45°$. The distance in cases where $\theta = 15°$ changes in y-direction to 0.14 m. The total number of time steps required for each case to reach the final positions are listed in Tab. 4.1.

Table 4.1: Time steps required for each case.

Courant number	**Coarse**	**Medium**	**Fine**
0.2	138	275	413
0.5	55	110	165

Since the exact solution is known, the solution error can be defined to qualify the different schemes. The same approach taken by Yu et al. (2001) is used to examine the convergence of each scheme towards the exact solution. The average error is

$$E_{av} = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{\alpha_i^{exact} - \alpha_i}{\alpha_i^{exact}} \right| \tag{4.2}$$

where N is the number of cells and α denotes the approximative solution. The exact solution is the mere translation of the step profile to the final location.

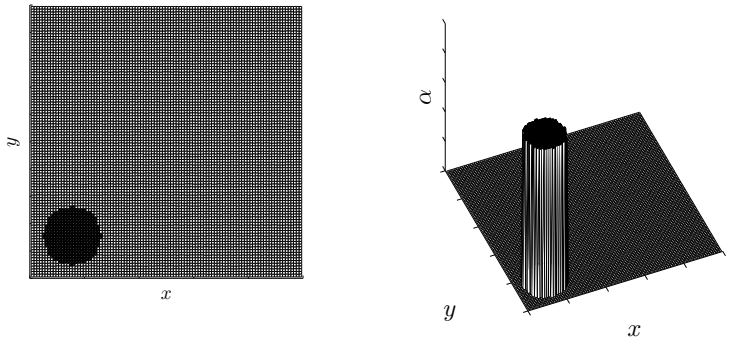


Figure 4.3: Circle-shaped step profile.

4.1.2 Results

The results from the convection of the step profile in an oblique velocity field are presented. The first figures depict one-dimensional cuts of the profiles along their centre lines. The cuts are taken in x-direction in the cases where $\theta = 45°$ and in y-direction when $\theta = 15°$, since otherwise the effects of the different angle are barely visible. The exact solution is denoted with the red line. Also the convergence of each scheme is studied via the average error defined by Eq. (4.2). The two-dimensional plots of each scheme and case are given in App. A, where the final position of the profile is shown together with the contours of $\alpha = [0.1, 0.5, 0.9]$.

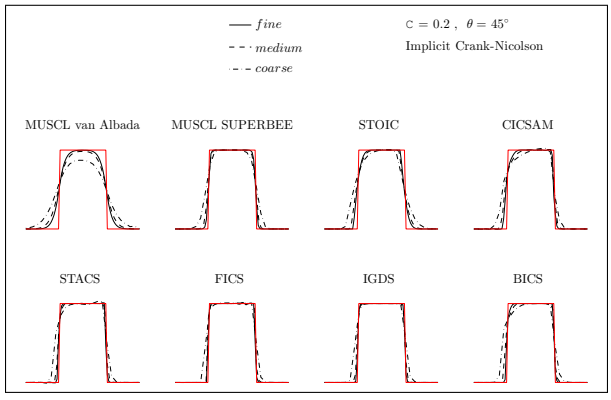


Figure 4.4: Centre lines of the square step profiles. Implicit time integration with $\mathsf{C} = 0.2$ and $\theta = 45°$

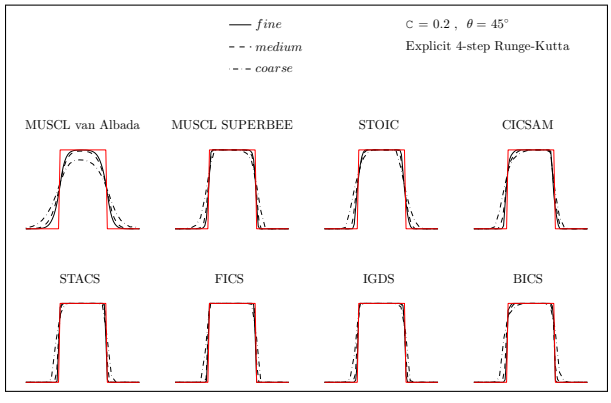


Figure 4.5: Centre lines of the square step profiles. Explicit time integration with $\mathsf{C} = 0.2$ and $\theta = 45°$

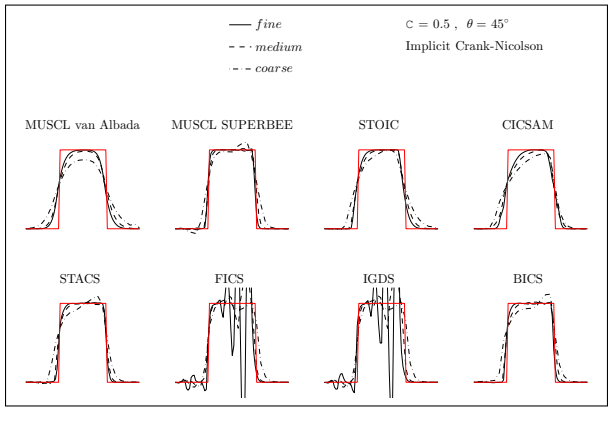


Figure 4.6: Centre lines of the square step profiles. Implicit time integration with $\mathsf{C} = 0.5$ and $\theta = 45°$

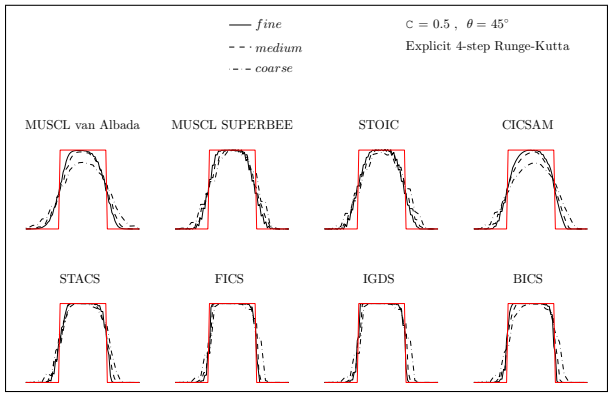


Figure 4.7: Centre lines of the square step profiles. Explicit time integration with $\mathsf{C} = 0.5$ and $\theta = 45°$

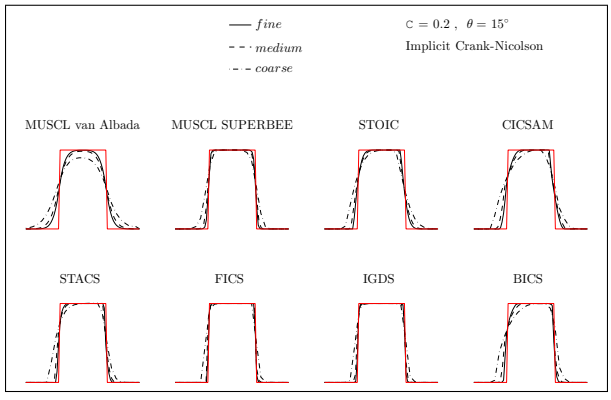


Figure 4.8: Centre lines of the square step profiles. Implicit time integration with $\mathsf{C} = 0.2$ and $\theta = 15°$

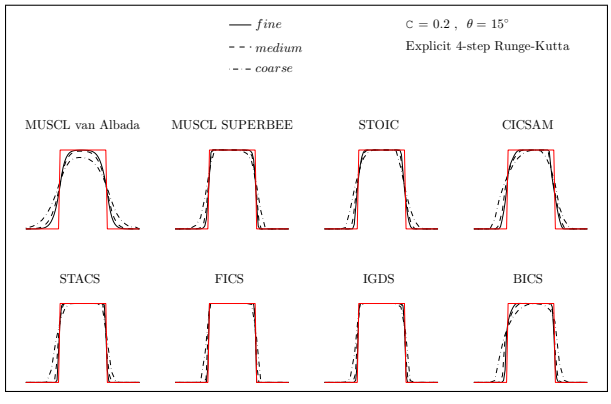


Figure 4.9: Centre lines of the square step profiles. Explicit time integration with $\mathsf{C} = 0.2$ and $\theta = 15°$

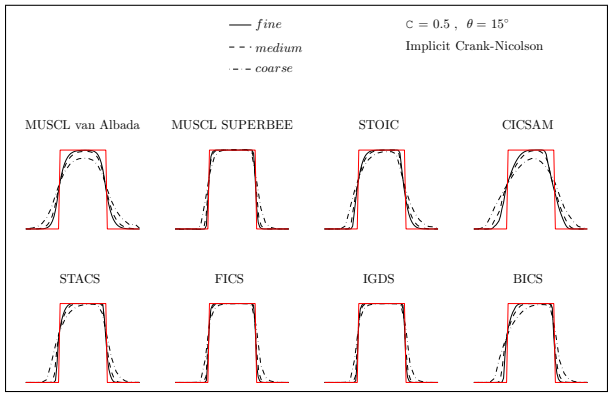


Figure 4.10: Centre lines of the square step profiles. Implicit time integration with $\mathsf{C} = 0.5$ and $\theta = 15°$

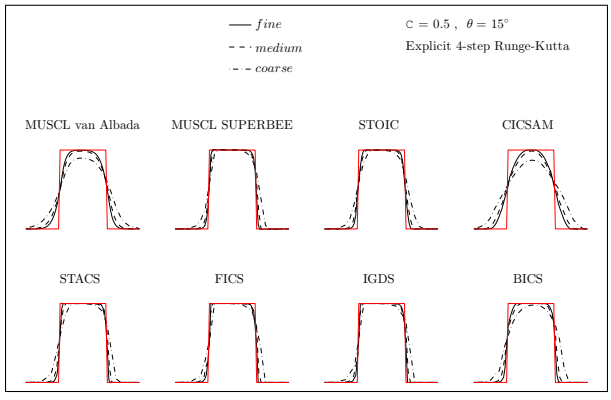


Figure 4.11: Centre lines of the square step profiles. Explicit time integration with $\mathsf{C} = 0.5$ and $\theta = 15°$

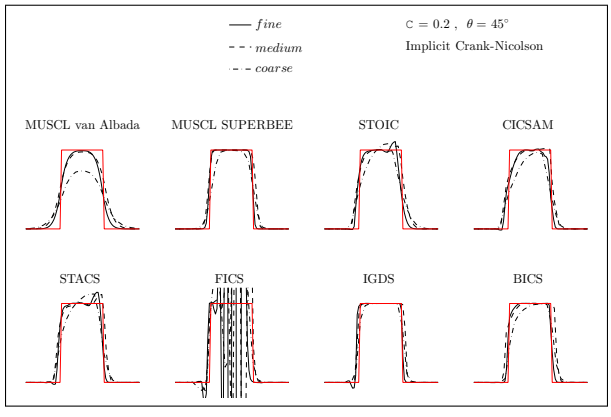


Figure 4.12: Centre lines of the circle step profiles. Implicit time integration with $\mathsf{C} = 0.2$ and $\theta = 45°$

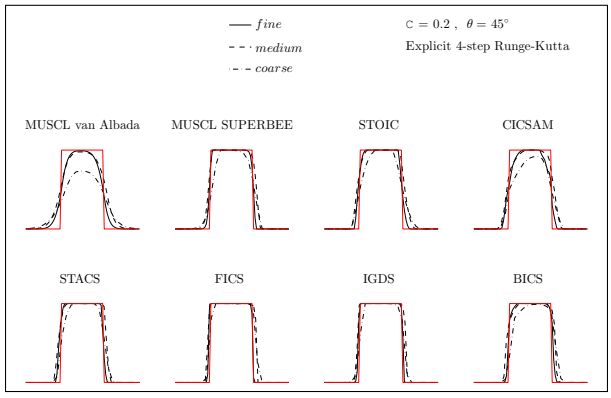


Figure 4.13: Centre lines of the circle step profiles. Explicit time integration with $\mathsf{C} = 0.2$ and $\theta = 45°$

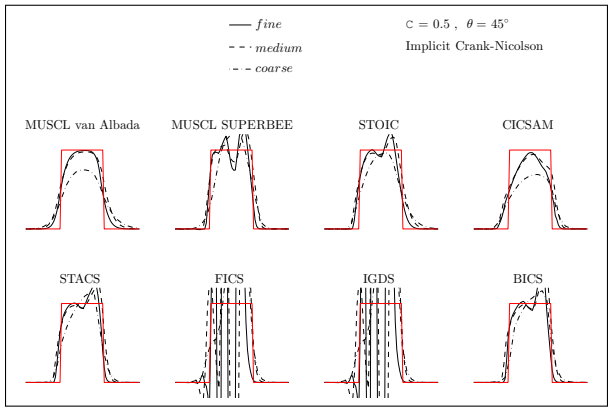


Figure 4.14: Centre lines of the circle step profiles. Implicit time integration with $\mathsf{C} = 0.5$ and $\theta = 45°$

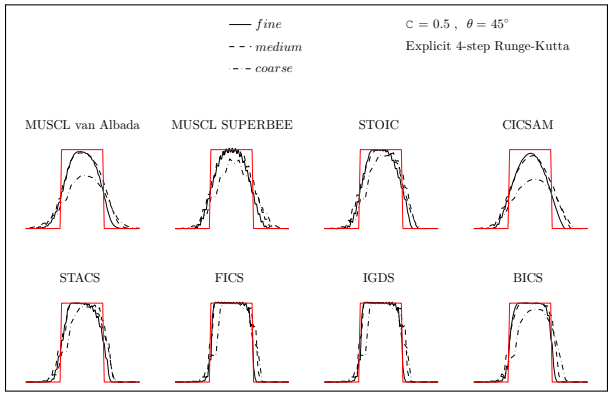


Figure 4.15: Centre lines of the circle step profiles. Explicit time integration with $\mathsf{C} = 0.5$ and $\theta = 45°$

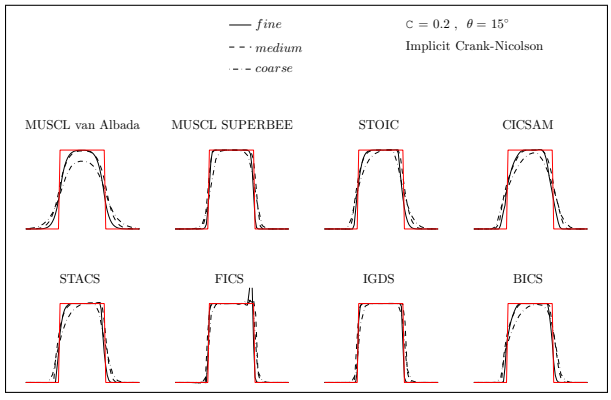


Figure 4.16: Centre lines of the circle step profiles. Implicit time integration with $\mathsf{C} = 0.2$ and $\theta = 15°$

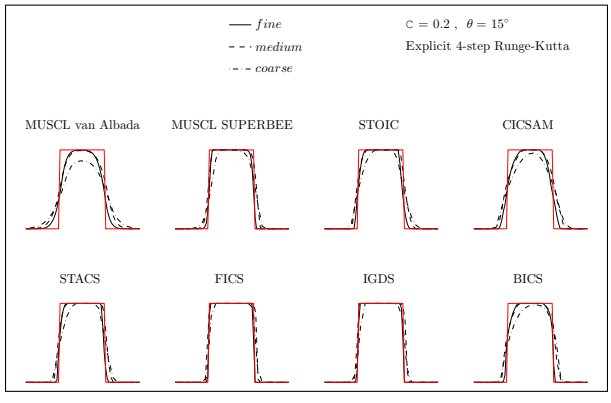


Figure 4.17: Centre lines of the circle step profiles. Explicit time integration with $\mathsf{C} = 0.2$ and $\theta = 15°$

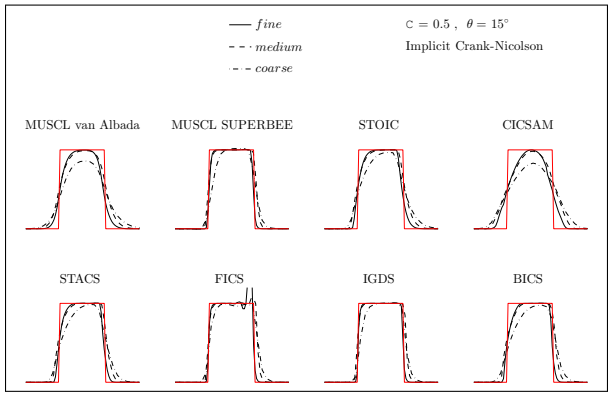


Figure 4.18: Centre lines of the circle step profiles. Implicit time integration with $\mathsf{C} = 0.5$ and $\theta = 15°$

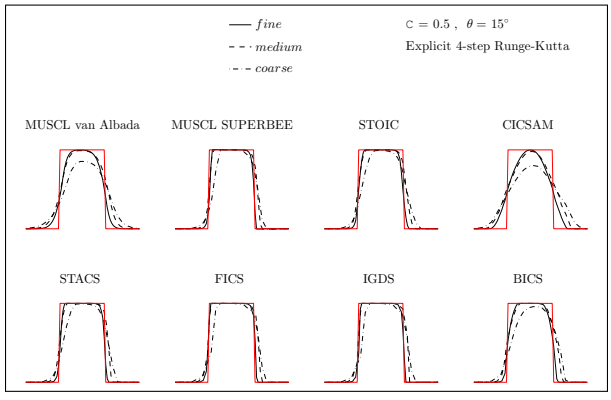
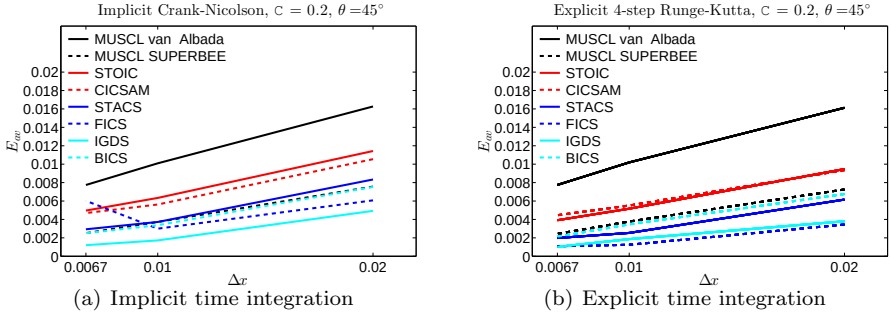
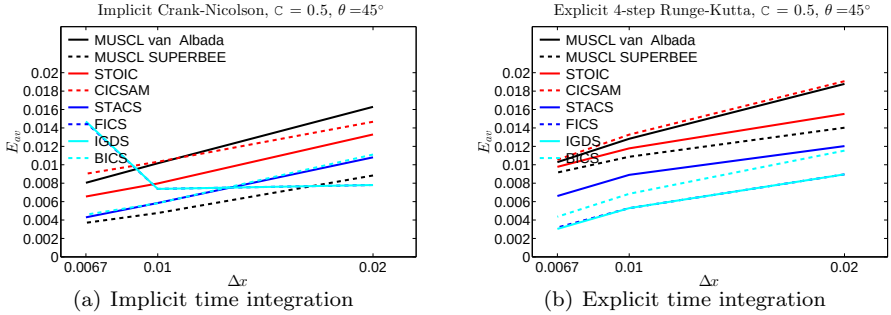


Figure 4.19: Centre lines of the circle step profiles. Explicit time integration with $\mathsf{C} = 0.5$ and $\theta = 15°$



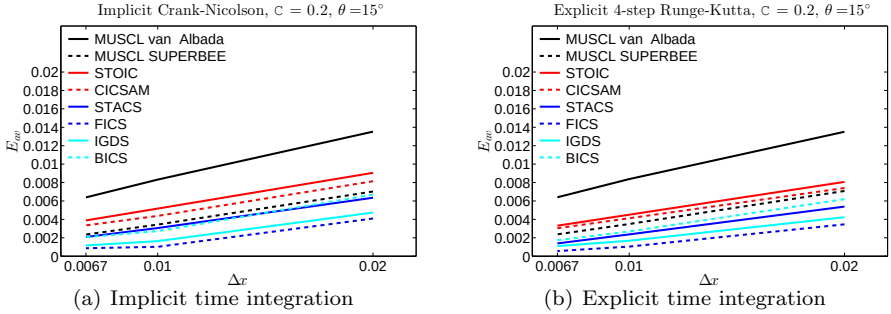
(a) Implicit time integration (b) Explicit time integration

Figure 4.20: Convergence of the approximate solution of the square shape towards the exact solution with different grid densities. The Courant number $\mathsf{C} = 0.2$ and the angle $\theta = 45°$



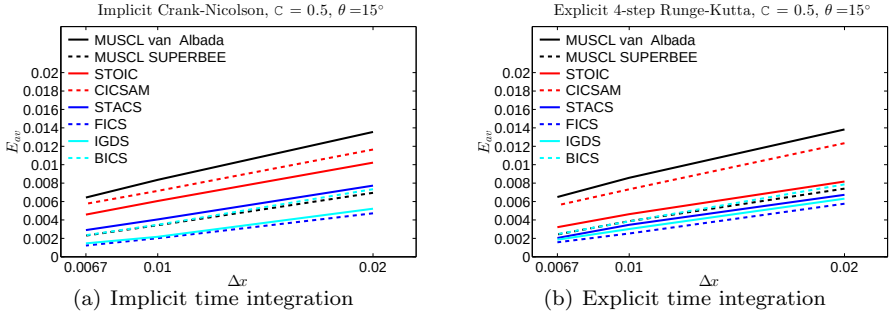
(a) Implicit time integration (b) Explicit time integration

Figure 4.21: Convergence of the approximate solution of the square shape towards the exact solution with different grid densities. The Courant number $\mathsf{C} = 0.5$ and the angle $\theta = 45°$



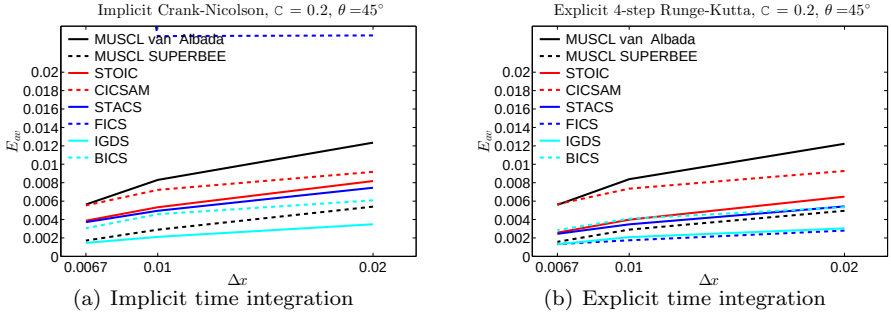
(a) Implicit time integration (b) Explicit time integration

Figure 4.22: Convergence of the approximate solution of the square shape towards the exact solution with different grid densities. The Courant number $\mathsf{C} = 0.2$ and the angle $\theta = 15°$



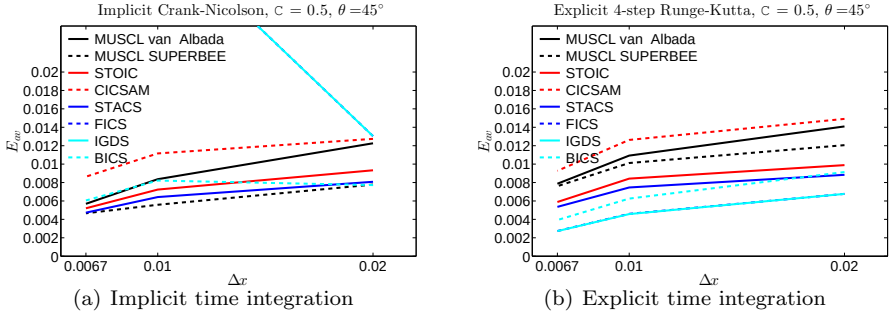
(a) Implicit time integration (b) Explicit time integration

Figure 4.23: Convergence of the approximate solution of the square shape towards the exact solution with different grid densities. The Courant number $\mathsf{C} = 0.5$ and the angle $\theta = 15°$



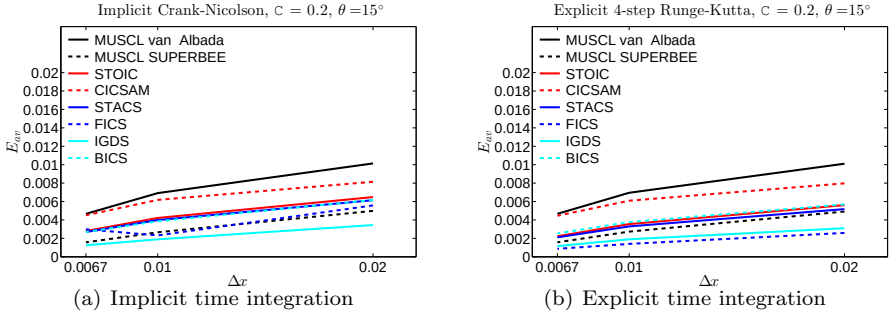
(a) Implicit time integration (b) Explicit time integration

Figure 4.24: Convergence of the approximate solution of the circle shape towards the exact solution with different grid densities. The Courant number $\mathsf{C} = 0.2$ and the angle $\theta = 45°$



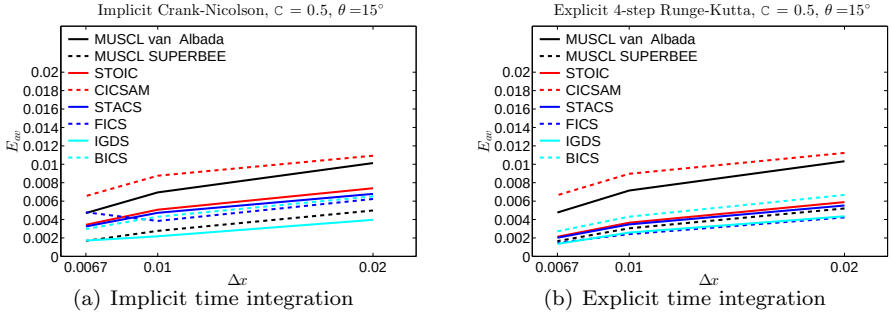
(a) Implicit time integration (b) Explicit time integration

Figure 4.25: Convergence of the approximate solution of the circle shape towards the exact solution with different grid densities. The Courant number $\mathsf{C} = 0.5$ and the angle $\theta = 45°$



(a) Implicit time integration (b) Explicit time integration

Figure 4.26: Convergence of the approximate solution of the circle shape towards the exact solution with different grid densities. The Courant number $\mathsf{C} = 0.2$ and the angle $\theta = 15°$



(a) Implicit time integration (b) Explicit time integration

Figure 4.27: Convergence of the approximate solution of the circle shape towards the exact solution with different grid densities. The Courant number $\mathsf{C} = 0.5$ and the angle $\theta = 15°$

4.1.3 Discussion

One of the most challenging individual problems in CFD was computed in this section. Many schemes especially devised for this were implemented in a two-dimensional computer code, and step profiles of two different shapes were considered. It can be shown that such schemes provide accurate and bounded results for one-dimensional situations, for example those that were considered in Sec. 2.2.1. Numerical experiments were also performed for all schemes presented in Sec. 3.3.2 in one-dimensional test cases, and the experience confirms the similar behaviour. The nature of the problem changes when multiple spatial dimensions are considered, since the flux can both enter a computational cell and leave from it over four faces in this case (cf. Fig. 3.7). The two-dimensional calculations were here reduced to a one-dimensional problem in each index direction since this provides no fundamental difficulty in structured meshes.

Overall, the implemented discretization schemes performed well. On average the discontinuity was captured within five computational cells on the finest grids, and the best result was within two cells. A clear majority of the computations maintained their physical reasonableness. These are significant steps considering the same formal accuracy is mostly maintained as was discussed in Sec. 2.2.1, and are due to the splendid accomplishments of the special numerical schemes.

Convection of the square profile

The results obtained with the smaller Courant number with the square profile and $\theta = 45°$ provide mostly excellent results for all schemes and profile shapes. Overall grid convergence is good, and especially on the finest grid level the solutions are very close to the exact one. The STACS scheme presents a small overshoot in the square case on the coarsest mesh using the implicit time integration. STOIC, CICSAM and BICS tend to smear the upper trailing edge of the profile a little with implicit time integration, but this does not spoil the result. The FICS presents an instability starting at the medium grid, and produces severe oscillations in the top right corner not visible in the one-dimensional plots, but only with the implicit integration. This disappears when using the explicit integration. The overshoots are

clearly visible in the convergence plots. It is difficult to distinguish the best scheme in this problem, since computations done with SUPERBEE limiter, IGDS, STACS and BICS give very sharp results. The latter two, however, suffer from some diffusion on the corners.

The results obtained with the higher Courant number with the square profile and $\theta = 45°$ show an increased difficulty. Most schemes present over- and undershoots on the coarsest grid using implicit time integration. These oscillations do not disappear when refining the grid for the FICS and IGDS, rather they explode such that the whole computations are meaningless. Other schemes stabilize upon refinement when the implicit integration is used, and provide good results on the finest grid. Notably the best are the SUPERBEE-limited scheme, STACS and BICS, when the van Albada-limited scheme, STOIC and CICSAM seem to suffer from excessive smearing. An interesting turn of events is observed when the four step explicit Runge-Kutta scheme is used. No violations on the physical bounds are produced by any schemes. However, the value of $\mathsf{C} = 0.5$ is strictly the upper bound for the stability of this scheme in two dimensions, and that seems to produce a checkerboard-like behaviour for all schemes in regions where $0 < \alpha < 1$; the compressive nature of these schemes can have an influence as well. Using the explicit integration the grid convergence of all schemes is good, and this time the best results are obtained using the FICS, IGDS and BICS schemes. This is interesting, since using the implicit integration, the former two broke down completely. It is also observed that the explicit integration provides more spreading to the SUPERBEE-limited, STOIC, CICSAM and STACS schemes.

The results obtained with the smaller Courant number with the square profile and $\theta = 15°$ repeats the behaviour of the corresponding case with the higher angle, with naturally less overall diffusion. Both explicit and implicit time integrations provide good results, and all computations stay in their physical bounds. The best results are obtained for both cases using the SUPERBEE-limited, FICS or IGDS schemes. The upper trailing edge of STOIC, CICSAM, STACS and BICS are mildly smeared, and grid refinement seems to minimize this smearing. The use of the higher Courant number introduces more diffusion in the profiles. Otherwise, the results obtained with the higher Courant number with the square profile and $\theta = 15°$ do not notably deviate from the previous observations.

Convection of the circle profile

The results obtained with the smaller Courant number with the circle profile and $\theta = 45°$ provide mixed results. Using the implicit integration, the SUPERBEE-limited and IGDS schemes show good grid convergence and very sharp resolution, but they seem to change the shape of the circle towards an oblique rectangle. In addition, the IGDS predicts a small undershoot at the trailing edge. Van Albada-limited schemes preserves the shape perhaps the best, but is very diffusive. Decent qualitative grid convergence is shown by CICSAM and BICS schemes in this sense. The FICS diverges and fails. Clearly the best results are obtained using the SUPERBEE-limited and IGDS discretization schemes when implicit time integration is used, but they do not preserve its shape very well on the finest grid. Again, the explicit time integration stabilizes the computations so that no physical bounds are violated. The SUPERBEE-limited, IGDS and FICS turn the circle into an oblique rectangle, whereas CICSAM and BICS distort it. The van Albada-limited computations preserve the circular shape, but still they are the most diffusive on all grids. The STACS and STOIC schemes give sharp results and also preserve the shape of the circle in the explicit computations better than the other schemes.

The results obtained with the larger Courant number with the circle profile and $\theta = 45°$ provide perhaps the most difficult case in this battery. Only van Albada-limited and CICSAM schemes maintain their physical bounds, but computations with the CICSAM scheme deteriorate the circular shape completely when the implicit integration is used. All other schemes fail, and present wavy behaviour of varying severity atop and in the wake of the profile. Yet again, the use of explicit time integration ensures the solution is bounded. Sharp results are obtained with the FICS and IGDS but the profile distorts into a hexagon. The BICS scheme gives decent results, but does not preserve the circular shape adequately. Notable, though relatively small, oscillations are still visible in the SUPERBEE-limited computations, and again the result is smeared.

The results obtained with the smaller Courant number with the circle profile and $\theta = 15°$ give good results compared to the previous case. Very good convergence and sharp approximations to the profile are obtained using the SUPERBEE-limited, STACS and IGDS schemes using both time inte-

gration methods. SUPERBEE-limited scheme and the slightly more diffuse STACS provide better shape of the profile than the IGDS, which turns toward an oblique square. The FICS scheme shows an overshoot at the upper leading edge of the circle profile when using implicit integration. The use of explicit integration removes the overshoot, and that scheme gives then very sharp results but with distortions to the shape. The STOIC, STACS and BICS schemes show a tiny increase in the diffusion of the upper leading edge when the explicit time integration is used, and hence the circular shape is better retained. The results obtained with the higher Courant number with the circle profile and $\theta = 15°$ do not notably deviate from these observations.

Effects of the shape of the profile

In this work, the circle shape profile provides means to study the shape preservation properties of the discretization. In the convection of the square shape, this is not so obvious since more compressive schemes retained the step-like front automatically using this grid configuration. Some schemes however found it difficult to produce good approximations to the sharp corners of the square where also the angle γ goes through an abrupt change. Once the more diffusive behaviour had stepped in, also the angle changed in a more continuous fashion and this sustained the diffusive process. In the cases involving the circle profile, the aforementioned squeezing was clearly visible, but this time rather obliquely to the grid lines. This was observable in the most compressive schemes.

In the computations with the square shape, the normal direction of the α distribution mostly aligns with the normal direction of the cell faces when a step is present. Therefore, the more compressive FICS, IGDS and BICS show somewhat similar behaviour, albeit only on the leading edges of the profile. The BICS maintains the compressive characteristic until higher angles γ as is shown in Fig. 4.28 but reverts to more diffusive GDS scheme. The computations using the circle profile enhance this diffusive tendency. The STACS returns to higher-order schemes earlier than the other that come with the blending, but the STOIC in itself is a combination of different discretizations, and comes with mildly more anti-diffusion than the GDS scheme. For this reason STACS is sharper at the trailing edge than the BICS or CICSAM, but more diffuse than the FICS. The latter observation

is due to the mentioned differences in the blending, and the FICS biases to a more compressive scheme on a wider range of angles γ, which is also visible in the notable similarities between the IGDS and FICS schemes. The CICSAM scheme has the same blending function as the FICS does, but it blends the HYPER-C with UQ. Especially with the higher Courant number it is very diffusive, as could be expected from its formulation. The circle case however proved to be very challenging, and most schemes distort its original shape. It seems that the STOIC scheme is very good for this, and the STACS mimics this behaviour successfully with the good shape preservation properties.

The convergence upon grid refinement has a steeper slope throughout the most computations when the square shape is considered. The absolute values however are slightly smaller from the circle shapes, albeit the case was clearly more challenging for most schemes. The form of the equation for the average error is relative such that it omits these types of effects. It alone will not give sufficient indication of one scheme over the other in this case, but combined with the observations on the convected shapes it provides good qualification on the characteristics of the schemes.

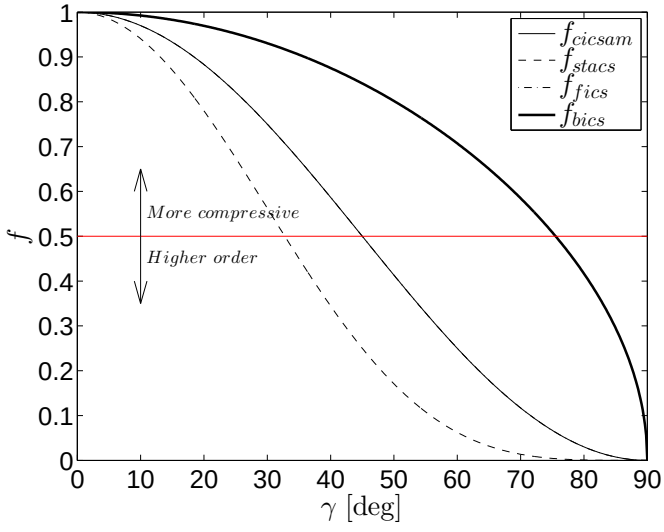


Figure 4.28: Blending functions of the CICSAM, STACS, FICS and BICS schemes. Note that those for CICSAM and FICS are the same

Effects of time integration

A rather curious result was that the explicit time integration showed the stabilizing properties in such a way that a computed solution remains within its physical bounds. That is, all schemes that were integrated with the explicit method, given that its own stability limits were not violated, remained within these bounds. Effort has been put on the spatial discretization of the convection term, but the standard higher-order time integration methods have been used here. Jasak (1996) showed that the simple implicit Euler integration, Eq. (3.103), produces a numerical diffusion comparable to that of first-order spatial discretization, whereas the explicit Euler integration, Eq. (3.102), produce similar anti-diffusion. The use of the second-order centred discretization in space is stabilized by such implicit treatment, for instance. Unfortunately, first-order methods with positive artificial diffusion smear the discontinuity into unrecognizably smooth humps on practical temporal and spatial grids. The Crank-Nicolson integration, which is an average of the explicit and implicit terms, Eq. (3.109), is just on the stability limit.

The pure convection equation does not contain any physical diffusion. Hence, for the solution algorithm to be stable, and for the solution to converge toward the weak solution of the conservation law, a non-negative amount of diffusion must follow from the numerical scheme. It has been shown to be necessary to employ compressive characteristics in the numerical approximation of the α equation (Ubbink, 1997), since it is supposed to be a step profile. Moreover, this has other consistency requirements (cf. Sec. 2.3). It was observed that the CBC in itself in the spatial discretization will not guarantee physically reasonable solutions, for certain two-dimensional cases considered here. Even the upper limit of the TVD seems not to suffice in the most severe case. The spatial discretization schemes that include the most negative diffusion obviously proved to be the most unreliable[1]. When viewing the case of $\mathsf{C} = 0.5$ and $\theta = 45°$ with the circle profile, all but the van Albada-limited (TVD) and the CICSAM scheme (NVF) fail when using implicit time integration, the best results provided by the former. The CICSAM scheme turns more diffusive as the Courant number is increased, and is additionally blended between very compressive and the most diffusive of the NV schemes used in this study. Yu et al. (2001), Gao et al. (2012) and Wei et al. (2003) discuss the applicability of the CBC, concluding that it is only

[1]Naturally no program is free of coding errors, but the results are consistent.

a sufficient condition. That is to say there may be other areas located in the NVD that produce bounded schemes. None of these were addressed in this study. This is due to the fact that the requirement of a discretization to be a second-order TVD scheme is more restrictive than any of these (Waterson and Deconinck, 2007), including the CBC. Leonard (1991) discusses that the NVF was motivated to overcome the inherent Courant number dependency of the TVD, to devise a less restrictive and more general approach for the boundedness of the convection discretization scheme.

These give some indication of two issues. Firstly, the four-step explicit Runge-Kutta time integration seems to contain enough positive diffusion to stabilize all computations such that the physical bounds are always preserved (with one exception, *i.e.* the same case as above using the SUPERBEE limiter, where one sees tiny wiggles over the value of 1). This reveals itself upon close inspection of the square profile case using $\mathsf{C} = 0.5$ with $\theta = 15°$. The implicit integration gives almost exact step profiles throughout the computations with the most compressive schemes, while the explicit integration introduces slightly more diffuse approximations in the direction of the higher Courant number. Similar effects can be seen in App. A. This behaviour might be enough to enforce a small positive diffusion coefficient in the modified equation, and provide a more stable algorithm. Further study is definitely needed. It is here noted that the Crank-Nicolson integration was sometimes used with tiny implicit bias, but tests proved it too diffusive to exercise in this study. This presents, however, an idea worth considering.

Secondly, as was mentioned many times before, both the TVD conditions and the CBC were initially formulated for one dimensional problems with explicit integration. Here, they are used in a one-dimensional fashion in both index directions on the cell faces. The flux calculation is first performed in the i-direction, and then in the j-direction, after which the residual is evaluated. Earlier in this thesis, one issue was presented with respect to the calculation of the fluxes in this way. It is obvious from certain results here that the two-dimensional implementation needs more caution, and cannot necessarily be implemented in this manner for the α equation. For example, Ubbink (1997) propose a complex predictor-corrector-type scheme to account for the multidimensional cases on arbitrary grids. Again, the use of the explicit time integration respected the physical bounds in all cases (with the above exception). In the literature the NVF has received wide attention, but

only a recent article by Moukalled and Darwish (2012) accounts for specific bounded time integration schemes.

Effects of grid resolution and Courant number

Nearly all studied cases and schemes present monotone convergence upon grid refinement. In a majority of cases, the relative average error is reduced below 0.002. This behaviour is also visible in the figures depicting the final shapes of the profiles. The use of the most compressive methods usually capture the discontinuity between two cells. This was clearly visible in most results obtained with FICS and IGDS schemes. Thus, upon grid convergence, the 'smearing' is not mitigated but merely the discontinuity is captured within the smaller cells, and the grid convergence is rather linear. In less compressive schemes, the numerical diffusion is a function of the cell size, and the grid convergence is not as dramatic.

The use of the smaller Courant number gave good results in all cases. Only the FICS failed when the circle profile was considered. Very sharp results were obtained with the smaller Courant number on the finest grids. Also the convergence upon grid refinement was better in the square case using the smaller Courant numbers. In the more challenging circle cases the Courant number had a more pronounced influence than on the square case. Few schemes that utilize the blending option turn to more diffusive ones as the Courant number is increased, which can be expected from their NVDs.

Behaviour of the different schemes

It is here noted that most schemes present very good resolution in the majority of cases, but some are better than the others in terms of consistency. A specific scheme does not immediately rise above all, nor does one plummet below the tolerance. Next, general remarks and observations on all schemes are provided.

The **van Albada**, as is expected, provided the most diffusive solutions to the convection problem. This is evident, *e.g.* from Fig. 3.16. Two alluring features, likely originating from this behaviour, were observed. Even in the most demanding cases, the shape of the profile was upheld, and the

approximations never came with non-physicalities. The scheme maintains a steady grid convergence in all cases. It is very close to linear in the square case and changes the slope toward a steeper one in the circle case upon grid refinement. Note that this scheme was also used in computation of the flow over a submerged ground elevation in the next section.

The **SUPERBEE** is a compressive limiter, and maintains the second-order TVD property. It is due to these facts that it proved one of the best methods for the problems studied here. Mostly the step discontinuity is resolved within four cells, and the performance is comparable with the most compressive NVFs. The convergence upon grid refinement is nearly monotone throughout, and often better than most schemes. This consistent behaviour is an asset amongst these methods. Note that this scheme was also used in computation of the flow over a submerged ground elevation in the next section.

The **STOIC** scheme is an early one based on the NVF. Qualitatively it provided good resolution in all cases. It is mainly a combination of central difference and QUICK schemes, and its compressive segment is rather short. Due to this, it provides slightly more smeared solutions than the other schemes. The convergence upon grid refinement is monotone throughout, and the scheme shows mediocre levels when compared to the other schemes in most cases considered.

The **CICSAM** is a blended discretization scheme, and it blends the HYPER-C and the UQ discretizations, but in such a way that it depends on the local Courant number. This provides good results in cases where $\mathsf{C} = 0.2$, whereas higher values increase numerical diffusion in all cases studied. The CICSAM, however, tends to distort the original shape of the square and circle profile toward a Siberian Flying Squirrel, by the looks of it; cf. Fig. A.7. Among the schemes that were studied, the overall convergence upon grid refinement is rather poor though monotone in all cases.

The **STACS** is a blended discretization scheme, and it blends the HYPER-C and the STOIC schemes. The latter proved to be a good one in itself. HYPER-C is the most compressive of all the schemes, and together these seem to give good resolution. Overall the results obtained using the STACS scheme are qualitatively among the top. Its convergence is in the within the best four schemes in comparison to the others. It is somewhat similar

to, though better than, that of the STOIC scheme. The scheme started the biasing toward the high-resolution scheme at smaller angles γ than the other blended schemes, and this seemed to be an asset.

The **FICS** is a blended discretization scheme, and it blends a highly compressive scheme in the parallel mode and one resembling the central difference in the normal mode. The difference between the two schemes, in terms of their compressive behaviour, is one of the steepest among all that were studied here. This type of blending provided ones of the sharpest results upon the cases studied here, and there are several ones in which the FICS gave the best results. The convergence upon grid refinement is amongst the two best in such cases. It, however, showed the most abrupt behaviour of all schemes, and completely exploded in few instances.

The **IGDS** is another early scheme based on the NVF. It features a quadratic profile in the NVD, and gives more compressive behaviour than the STOIC, for instance. The NVD is quite close to that of the FICS scheme in cases where the normal of the α distribution aligns with the normal of the cell faces, and this explains the myriad of similarities that are observable with these schemes. The overall performance of the IGDS is likewise very good, and it is among the best schemes in terms of resolution. It is observed that the scheme does not diverge as easily as the FICS does. The convergence upon grid refinement is among the two best in cases where the physical bounds are not violated. Note that this scheme was also used in computation of the flow over a submerged ground elevation in the next section. It is here pointed out that Jasak and Weller (1995) give a recommendation of a limit of $\mathsf{C} < 1/3$ in multidimensional cases, but it was here observed that in many, but not in every, cases this posed no problems. This also seems to be the first time the scheme is reportedly applied with an explicit time integration, and it showed very good results for the convection problem.

The **BICS** is a blended discretization scheme, this time blending between a quadratic and the GDS schemes. The quadratic one resembles closely the IGDS but it depends slightly on the local Courant number. It is observed that this type of behaviour showed a stabilizing effect in cases where the IGDS failed, and naturally otherwise yielded a bit more diffusive approximations than the IGDS. The BICS maintain the compressive properties for higher angles θ than any other blended scheme (Fig. 4.28). The BICS,

however, tends to distort the original shape of the square and circle profile toward a Siberian Flying Squirrel, by the looks of it; cf. Fig. A.4. This behaviour is much akin to the CICSAM scheme, and this comes in varying severity throughout the battery. The resemblance to CICSAM is not a surprise, since the NVDs of these schemes are quite similar in this respect. These effects are then again slightly more serious with the CICSAM scheme. Be that as it may, the convergence upon grid refinement appears within the top four.

In conclusion, based on accuracy, consistency and the resolution of the α distribution, the best out of the eight studied here is the SUPERBEE-limited MUSCL differencing scheme. It gives non-physical results in only one case (the van Albada limiter and CICSAM give none), and otherwise shows very sharp resolution with good grid convergence and the shape preservation properties. It is most evident that a compressive method is necessary, and a more compressive scheme than the SUPERBEE, like the FICS, IGDS or STACS, will provide better resolution, but with the observed limitations. The best resolution obtained with the SUPERBEE-limited scheme was capturing the discontinuity within four cells, whereas with the more compressive FICS, IGDS, STACS and BICS this was two cells. The idea of blending the schemes is definitely worthwhile considering in the future. It goes without saying that one needs to compare these also using an actual flow problem, in sync with a Navier-Stokes solver.

4.2 Computation of flow over a submerged ground elevation

This section presents the computation of flow over a ground elevation submerged in water, with the computational domain extending to the gas above the water. The aim of this section is to assess the capabilities of the homogeneous mixture model of FINFLO against the code YAFFA that has a traditional VOF method implemented. It also serves as aid to recognize possible future issues within this approach for this thesis serves as an intermediate stage for the development process of FINFLO. The bump case was chosen for this purpose due to the simple geometry that was involved, and the intuitive exclusion of any trouble that may arise from, *e.g.*, a body in the flow field or an object piercing the interface. Additionally, the reference results were rather consistent and straightforward. Then, to get acquainted with the capabilities of the model, the parameters affecting the approximation of the forming wave, the stability and convergence properties as well as the applicability of the solution algorithm itself, and also due to the nature of the stage, only the sub-critical case is considered in this work.

4.2.1 Problem description

The flow of water over a ground elevation submerged to a finite depth is a straightforward test case for flow solvers with free surface resolving capability. Here, flow over and underwater elevation of Gaussian shape is chosen (Bet et al., 1996; Mikkola, 1999, 2009). It is a problem with simple geometry and features only two parameters that affect the physics of the flow. The flow is dominated mainly by pressure and inertial forces, so the deficiency in turbulence modelling bears no hindering effect at present. The steepness of the bump is rather mild, so no flow separation is expected to occur with the velocities used. In addition, the reference computations by Mikkola (1999, 2009) have been performed with the same codes as used in this work but with a different free surface solution method. Bet et al. (1996) used both a deforming grid and a level-set method to compute the case. Additionally, in this work, the present case is computed with the current interface tracking model of FINFLO as well. Details of that can be found in the work of

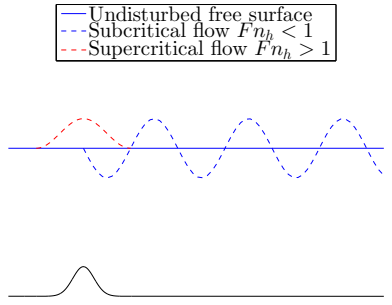


Figure 4.29: Examples of the types of the free surface deformations in subcritical and supercritical cases. Flow is from the left

Meinander et al. (2006). Similar parameters as were used by Mikkola (1999) are chosen here.

The ground elevation is of Gaussian shape and given by the expression

$$z = z_0 + A \exp(-\beta^2 (x - x_0)^2) \ , \tag{4.3}$$

where $z_0 = -1$ m, $A = h - h_b$, $\beta = 1$ and $x_0 = 0$ are chosen to match the reference computations. The present case corresponds to a so-called subcritical situation, *i.e.*, the depth-based Froude number $Fn_h = 0.567 < 1$ (Lowery and Liapis, 1999). Distinction of the cases are depicted in Fig. 4.29. The grid constructed for the computations extends from -30 m to $+30$ m in x-direction and from -1 m to $+5$ m in z-direction.

4.2.2 Initial and boundary conditions

A schematic view of the case is given in Fig. 4.30 with the initial distribution of the void fraction α. Acceleration due to gravity is in the negative z-direction, *i.e.* $\vec{g} = (0, 0, -1)^{\mathsf{T}} \cdot g$, where $g = 9.81$ m/s^2 is its magnitude. The inflow velocity $\vec{V}_{in}$ is chosen such that $Fn_b = |\vec{V}_{in}|/\sqrt{gh_b} = 0.567$. In YAFFA, the free stream boundary condition is set based on flow direction. In the inflow boundaries, velocity is set to the pre-described value $\vec{V}_{in}$ and the pressure is linearly extrapolated from the domain. The volume fraction is copied from the domain. In the case of the outflow boundary, pressure is set to the hydrostatic value and the velocity is linearly extrapolated from the domain. The void fraction is copied from the domain. In FINFLO, the

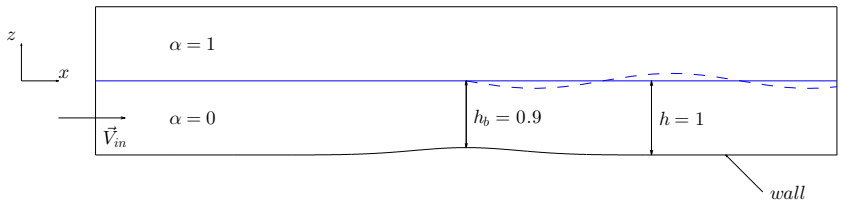


Figure 4.30: A schematic view of the submerged ground elevation. α denotes the gas volume fraction. The solid blue line depicts the initial free surface level and the dashed one gives a hint on the formation of the wave downstream of the bump.

free stream (external) boundary condition is set based on flow direction. In the inflow boundaries, velocity is set to the pre-described value $\vec{V}_{in}$ and the pressure is extrapolated. The volume fraction is set according to the initial still water level. In outflow boundaries, the pressure is set to the hydrostatic value based on the initial still water level, and velocity is extrapolated from the domain. The volume fraction is set according to the initial still water level. The boundary conditions are given in Tab. 4.2.

Table 4.2: Boundary conditions (BC) used with the flow solvers

Boundary	**BC for** YAFFA	**BC for** FINFLO
Inlet	Free stream	External
Lower wall	Slip	No-slip
Outlet	Free stream	External
Upper boundary	Mirror	Mirror

4.2.3 Computational conditions

For both flow solvers, the computational grid was created using the commercial grid generation tool Numeca IGG. A grid format conversion tool was written for converting the structured `PLOT3D` format to the unstructured one used in YAFFA. The two-dimensional computational domain corresponding to Fig. 4.30 is set to $x = [-30, 30]$ m and $z = [-1, 5]$ m with the centre of the bump located at $(0, -0.9)$ m. The domain is stretched so in order to seek low sensitivity to far-field boundary conditions. FINFLO uses multi-block structured grids, *i.e.*, problems can be run in parallel. Two grids, one finer and one coarser, were constructed for the computations with FINFLO. The

effects of grid resolution, clustering and cell sizes near the free surface are investigated with the two different types of grids. Results with the program YAFFA are reported from only one grid.

FINFLO is a three-dimensional RANS code, obliging one to use three-dimensional computational grids as well as a sufficient grid for the boundary layer. The code uses small Reynolds number models, hence the grid needs appropriate resolution in the boundary layer. A correlation from White (2006) for the skin friction coefficient is

$$C_f = \frac{\tau_w}{\frac{1}{2}\rho V_{in}^2} \approx \frac{0.027}{Re_x^{1/7}} \ , \tag{4.4}$$

where $Re_x = \rho V_{in} x/\mu$ is the Reynolds number. Now remembering the definition of friction velocity

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \tag{4.5}$$

and the dimensionless wall distance

$$z^+ = \frac{z u_\tau}{\nu} \ , \tag{4.6}$$

where z is the perpendicular distance from the wall, one can write for the first cell

$$z = \frac{z^+ \nu}{\sqrt{\frac{C_f V_{in}^2}{2}}} \ , \tag{4.7}$$

one may compute the first cell height $z = 1 \times 10^{-5}$ m for which $z^+ \approx 1$. The code YAFFA allows the use of significantly coarser grids in the vicinity of the wall, the main goal being the sufficient resolving of the disturbance created by the bump.

The computed, steady-state interface is to resemble that of Fig. 4.31. The wave length is $\lambda/h_0 \approx 2$ with the approximate maximum amplitude $A/h_0 \approx 0.1$. In this study, a resolution consisting of at least 30 computational cells for both is assumed to suffice. This sets the limits around the interface

$$\Delta x/h_0 \leq 0.067 \ , \text{ and} \tag{4.8}$$

$$\Delta z/h_0 \leq 0.0033 \ . \tag{4.9}$$

Relevant details of the grids used with FINFLO are gathered in Tab. 4.3.

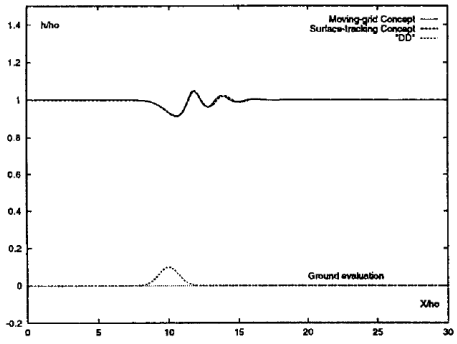


Figure 4.31: Computed free surface profile by Bet et al. (1996). h_0 is the height of the undisturbed free surface

Table 4.3: Details of the grids used with FINFLO

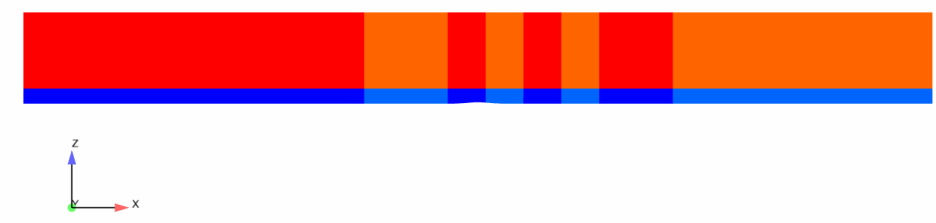
Grid	Detail	Description
Coarse	Number of cells	745 472
Fine	Number of cells	2 981 888
Coarse	Clustering in x-direction	4 blocks asymmetrically around the bump with $\Delta x = 44.9$ mm
Fine	Clustering in x-direction	4 blocks symmetrically around the bump with $\Delta x = 8.9$ mm
Coarse	Clustering in z-direction	60 cells around the free surface with $\Delta z = 3.3$ mm
Fine	Clustering in z-direction	60 cells around the free surface with $\Delta z = 3.3$ mm

These limits are not followed with the reference code, and the reason will be given later. The grid constructed for YAFFA has the size of 500×150 cells in x and z-directions, respectively, and 75 000 cells in total. The grid spans as $x = [-30, 30]$ m and $z = [-1, 1]$ m, and it is uniform throughout the domain.

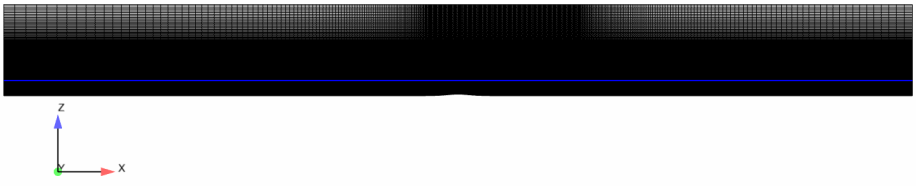
The coarser grid for FINFLO consists of 16 blocks. Each block has the size of $56 \times 4 \times 208$ cells in x, y and z-coordinate directions, respectively. The whole grid has 745 472 cells in total. The grid spans as $x = [-30, 30]$ m, $y = [0, 10]$ m and $z = [-1, 5]$ m. The grid is divided so that eight blocks are initially filled with liquid phase and lay beneath the $z = 0$ level, *i.e.*, the initial free surface level. The other eight blocks are on top of the $z = 0$ level and are initially filled with the gas phase. The block division is depicted in Fig. 4.32(a) and an overall view is given in Fig. 4.32(b). Clustering of the grid points near the free surface in z-direction is done such that there are 60 cells with constant spacing of $\Delta z = 3.3$ mm around the initial free surface. Outside this range, the grid is gradually coarsened with approximately the factor of 1.05, and again clustered for the boundary layer in the bottom blocks. Four blocks with constant spacing of $\Delta x = 44.6$ mm are located

asymmetrically around the bump in both below and above the $z = 0$ level; this exceeds the requirement by a factor of 1.5. The first of these begins at $x = -2$ m and the last ends at $x = 8$ m. Their length is 2.5 m each. The outer blocks are gradually coarsened towards the ends of the computational domain with similar expansion factor as was used in the z-direction. This grid was also used with the interface tracking model of FINFLO.

The finer grid for FINFLO consists of 16 blocks. Each block has $224 \times 4 \times 208$ cells in x, y and z-coordinate directions, respectively. The whole grid has 2 981 888 cells in total. The grid spans as $x = [-30, 30]$ m, $y = [0, 10]$ m and $z = [-1, 5]$ m. The grid is divided so that eight blocks are initially filled with liquid phase and lay beneath the $z = 0$ level, *i.e.* the initial free surface level. The other eight blocks are on top of the $z = 0$ level and are initially filled with the gas phase. Clustering of grid points is done in the vicinity of the bump by blocks, and the grid is symmetric around the bump. The block division is depicted in Fig. 4.34(a) and an overall view is given in Fig. 4.34(b). Clustering of the grid points near the free surface in z-direction is done such that there are 60 cells with constant spacing of $\Delta z = 3.3$ mm around the initial free surface. Outside this range, the grid is gradually coarsened with approximately the factor of 1.05, and again clustered for the boundary layer in the bottom blocks. There are two blocks directly before and after the bump in both below and above the $z = 0$ level with uniform grid spacing in x-direction, with $\Delta x = 8.9$ mm; this exceeds the requirement over seven times. Their length is 2 m each. These are followed by two blocks with length of 6 m where grid points are clustered towards the denser blocks. The first and last blocks in both above and below the $z = 0$ level are coarsened towards the ends of the computational domain, and they are 20 m each, again with clustering towards the denser blocks.

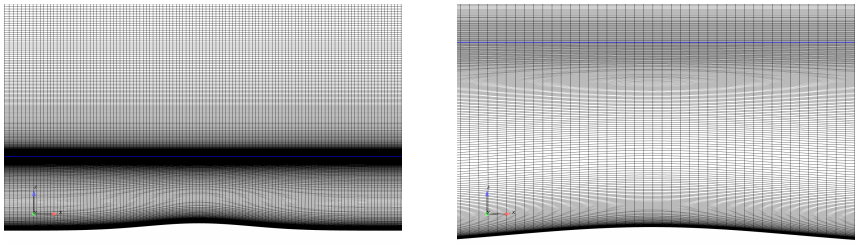


(a) Overview of the block division of the coarser grid. The blocks initially above the free surface are coloured with red, and those initially below it are coloured with blue. The different shadings distinguish the blocks.



(b) Overview of the coarser grid. The blue line denotes the undisturbed free surface level.

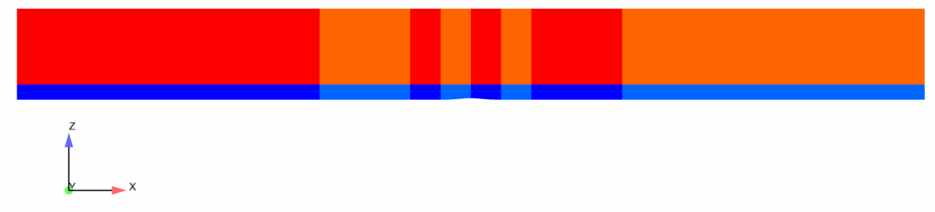
Figure 4.32: Overall views of the coarser grid.



(a) Overview near the bump.

(b) Closer view near the bump.

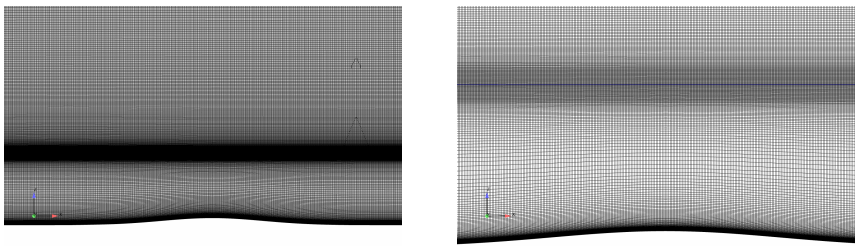
Figure 4.33: Details of the coarser grid near the bump. The initial free surface level is coloured with blue.



(a) Overview of the block division of the finer grid. The blocks initially above the free surface are coloured with red, and those initially below it are coloured with blue. The different shadings distinguish the blocks.

(b) Overview of the finer grid. The blue line denotes the undisturbed free surface level.

Figure 4.34: Overall views of the finer grid.



(a) Overview near the bump.

(b) Closer view near the bump.

Figure 4.35: Details of the finer grid near the bump. The initial free surface level is coloured with blue.

4.2.4 Results

The results from the simulation of two-dimensional flow over the submerged ground elevation are presented. All simulations with the mixture model of FINFLO are presented from three grid levels, and with the interface tracking model of FINFLO using one grid level. The computations were initiated on the third grid level of both grids, that is, every index direction has a third of the nodes of the original grid. Once those computations had reached a steady state, the results were interpolated to the finer grid level and the same procedure was repeated. Computations with the mixture model of FINFLO are performed on six different grids in total, and the reference computations using YAFFA are done on one grid. This section presents the formation of the interface between the gas and liquid as well as relevant plots of different pressure and velocity distributions, and convergence curves of various quantities. The evolution of the interface is studied from several iteration cycles, and the performance of the different discretizations for the void fraction is studied. Most focus is put on the results obtained from FINFLO.

The gas volume fraction distribution resolved with YAFFA is given in Fig. 4.36, and the distribution of the pressure difference is given in Fig. 4.37. The distribution of the pressure difference predicted with the interface tracking model of FINFLO is given in Fig. 4.38. The gas volume fraction distribution resolved with FINFLO are given in Fig. 4.39 and Fig. 4.40 for the coarse and fine grids, respectively. The corresponding distributions of the pressure difference are given in Fig. 4.41 and Fig. 4.42. Predictions of the free surface shape from all grid levels of the both grids are shown in Fig. 4.43. Comparison of the free surface shape from the reference results and those obtained from present computations is plotted in Fig. 4.44. More detailed plots of the pressure difference, total pressure and speed along various cuts in the x-direction are given in Fig. 4.45. Closer views of the distributions of the total pressure, which are given from several different locations along the interface, are gathered in Apps. B.1 and B.2 from computations with FINFLO, and in App. B.3 from computations with YAFFA. Most conclusions are drawn from the first grid levels of the two grids.

The computations started with the coarser grid using enough relaxation such that a solution could be obtained. Once it was established that the problem is solvable, a preliminary study of the parameters affecting the

computations was made. These included the method of extrapolation of the convective term in the bulk flow equations, the under relaxations for the pressure and velocity and the Courant number. Additionally, the effects of grid density and the discretization method used for the convection of the void fraction was included in these investigations. Observations will be presented for the two first, whilst more quantitative information is provided for the rest. And since the model is supposed to solve the free surface problem, investigation is done to see whether this is the case.

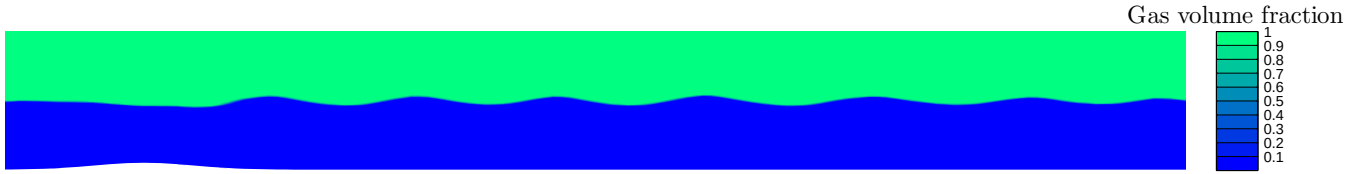


Figure 4.36: Distribution of the gas volume fraction from computations with YAFFA. The horizontal coordinate range is $x = [-2.0, 15.0]$ m and the vertical is $z = [-1.0, 1.0]$ m.

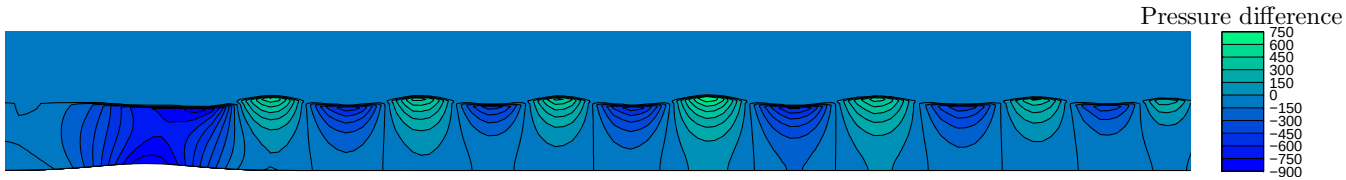


Figure 4.37: Distribution of the pressure difference from computations with YAFFA. The horizontal coordinate range is $x = [-2.0, 15.0]$ m and the vertical is $z = [-1.0, 1.0]$ m.

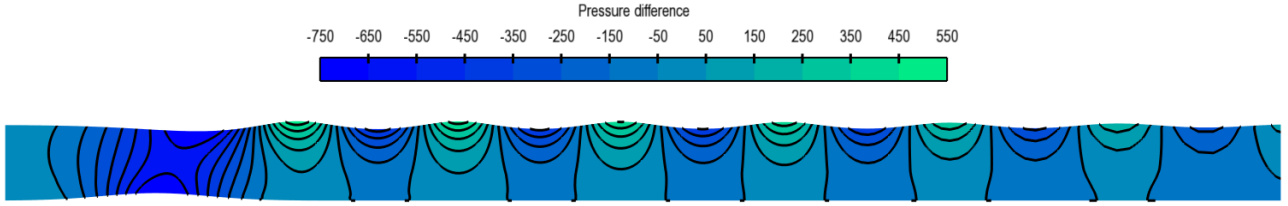


Figure 4.38: Distribution of the pressure difference from computations with FINFLO and the moving grid. The horizontal coordinate range is $x = [-2.0, 15.0]$ m.

Figure 4.39: Distributions of the gas volume fraction from computations with FINFLO using the three grid levels of the coarser grid. The first (finest) grid level is the topmost figure. The horizontal coordinate range is $x = [-2.0, 15.0]$ m and the vertical is $z = [-1.0, 1.0]$ m.

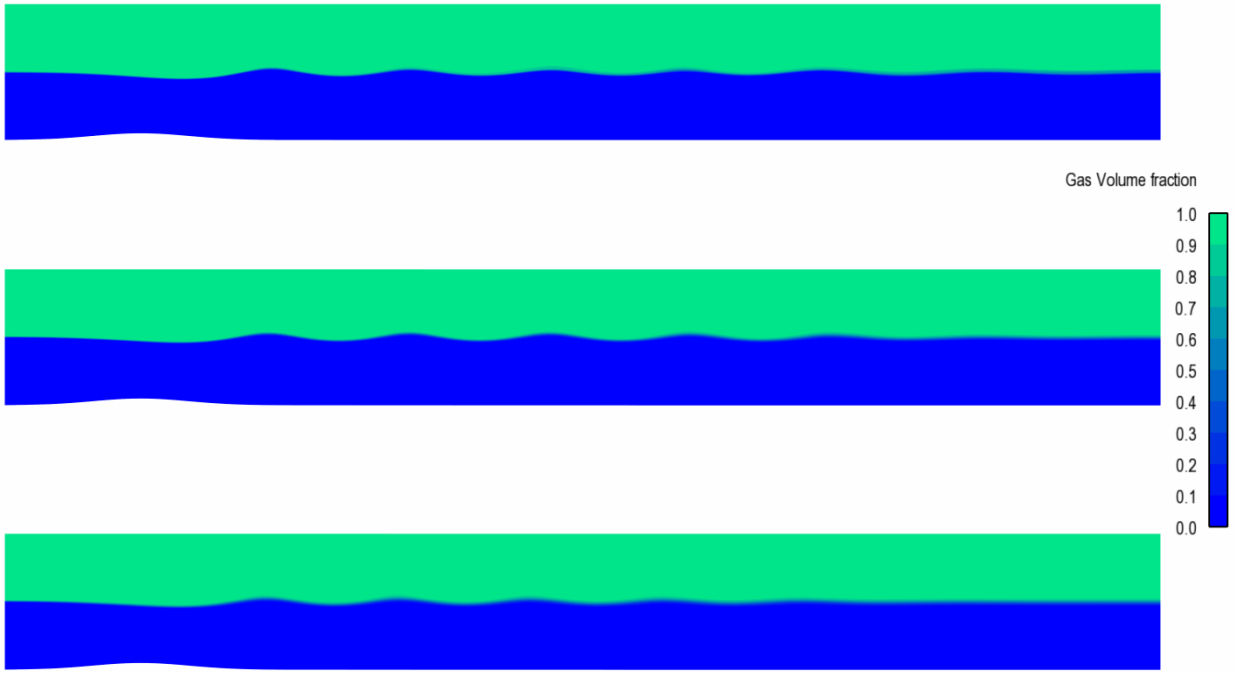


Figure 4.40: Distributions of the gas volume fraction from computations with FINFLO using the three grid levels of the finer grid. The first (finest) grid level is the topmost figure. The horizontal coordinate range is $x = [-2.0, 15.0]$ m and the vertical is $z = [-1.0, 1.0]$ m.

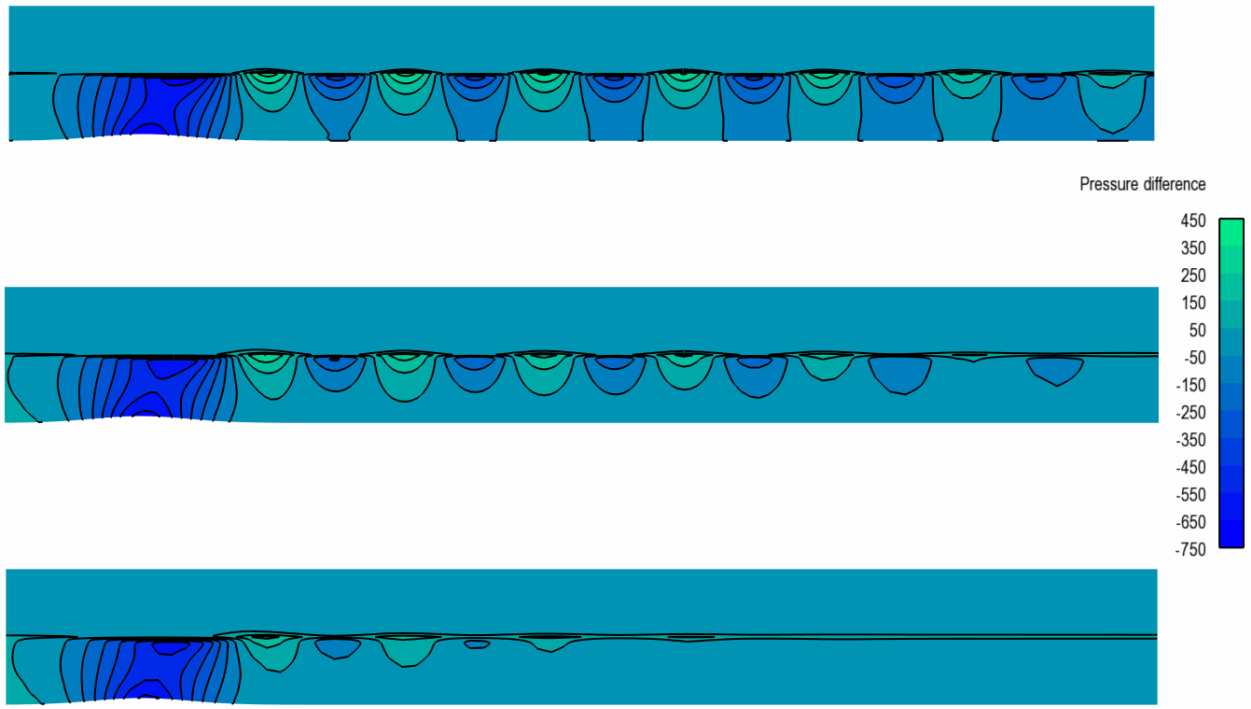


Figure 4.41: Distribution of the pressure difference from computations with FINFLO using the three grid levels of the coarser grid. The first (finest) grid level is the topmost figure.The horizontal coordinate range is $x = [-2.0, 15.0]$ m and the vertical is $z = [-1.0, 1.0]$ m.

Figure 4.42: Distribution of the pressure difference from computations with FINFLO using the three grid levels of the finer grid. The first (finest) grid level is the topmost figure. The horizontal coordinate range is $x = [-2.0, 15.0]$ m and the vertical is $z = [-1.0, 1.0]$ m.

4.2.5 Discussion

This section presents the first reported computations with the homogeneous mixture model of FINFLO to capture the free surface. These are an extension to those the author readily presented at a recent Numerical Towing Tank Symposium (Viitanen, 2014). The model is mostly used as it is, and the mass transfer is disabled along with some auxiliary relations of the material parameters that are dependent on the cavitation number. In the beginning there were lots of headaches, numerous trouble-shootings, test calculations and try-outs to determine if the model is indeed suitable for solving the problem at hand. Revision of the solution algorithm presented earlier inspired hope into this phase. Piece by piece the puzzle was placed together, and the results and some experiences can be summarized here. By observing Figs. 4.36, 4.37, 4.38, 4.39, 4.40, 4.41 and 4.42 it is seen that the results are very promising in comparison to the reference solutions.

The distributions of the pressure difference are qualitatively very similar, differing mainly in magnitude. YAFFA predicts higher positive pressure differences at the wave crests, Fig. 4.37, whereas the FINFLO gives slightly greater negative pressures near the bump on the finer grid, Fig. 4.42, but the negative pressures are very close to those obtained with YAFFA in the wave troughs. The computations with the interface tracking method of FINFLO provides similar distributions of the pressure difference, given in Fig. 4.38, as was predicted with the coarser grid, Fig. 4.41, which is expected since the initial grids used were the same. However, the interface tracking method gives slightly higher values of the positive pressure differences at the wave crests, being approximately the same as were obtained with the finer grid. The results from all grid levels on both grids, in terms of the contour $\alpha = 0.5$, are plotted in Fig. 4.43. The computations with the coarser grid show very good results, whereas the finer grid shows increased phase difference and slightly inconsistent convergence. The plots show also the results obtained from the SUPERBEE-limited discretization for the volume fraction; this is discussed in more detail later.

The comparison of the present computations with the reference results are plotted in Fig. 4.44. The results show satisfactory agreement in terms of the contour of $\alpha = 0.5$ using the interface capturing methods, and the actual free surface that is obtained from the interface tracking methods.

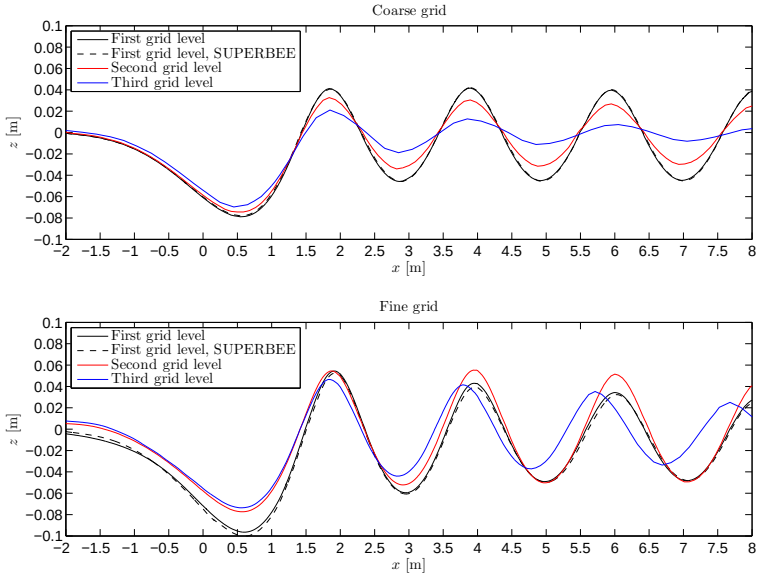


Figure 4.43: Comparison of the free surface shapes of computations from FINFLO on the coarse and fine grids. The computed shapes are in terms of the contour of $\alpha = 0.5$.

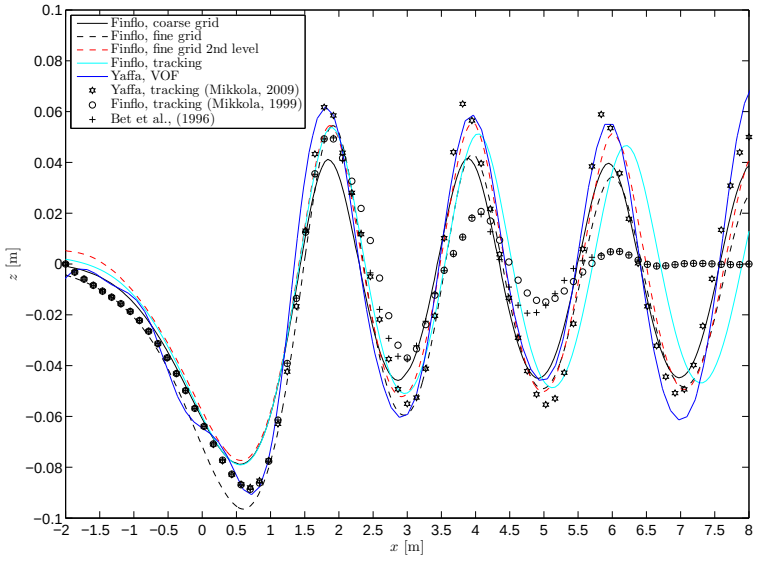
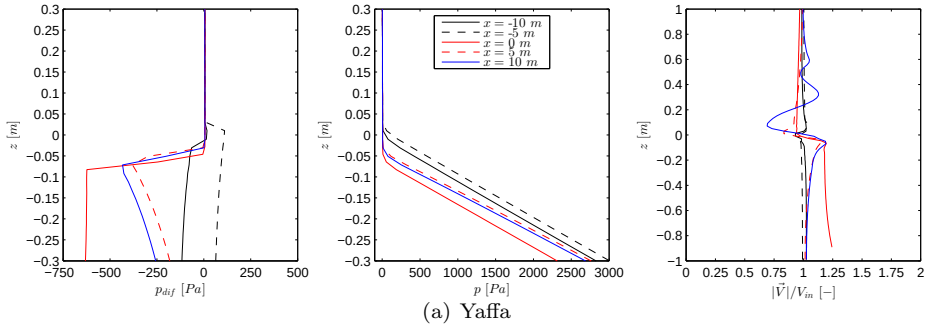
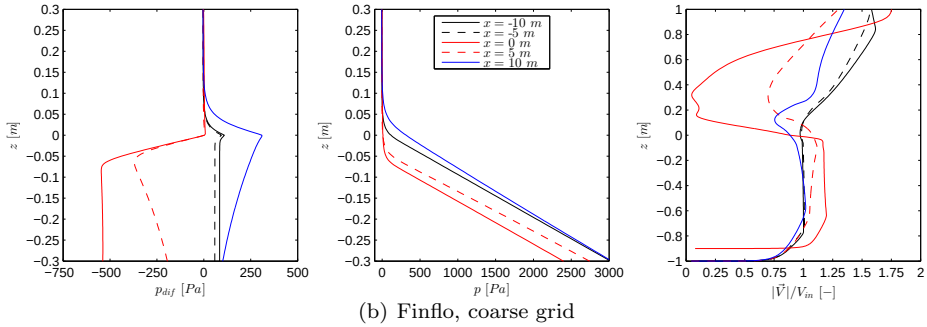


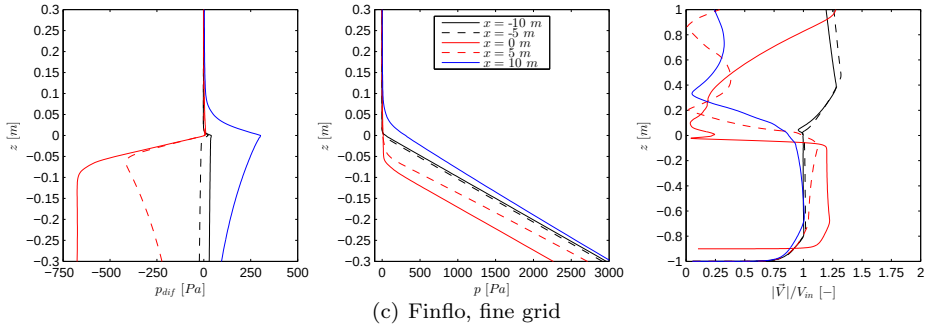
Figure 4.44: Comparison of the free surface shapes of computations form YAFFA and from FINFLO to the reference computations. Both results from the mixture model of FINFLO are van Albada-limited. The shapes computed with the capturing models are in terms of the contour of $\alpha = 0.5$.



(a) Yaffa



(b) Finflo, coarse grid



(c) Finflo, fine grid

Figure 4.45: Distributions of pressure difference, pressure ($= p_{dif} - \rho g z$) and non-dimensional speed on several cuts along the x-axis. The constant ambient pressure p_∞ has been subtracted from the pressure calculated with FINFLO. Note the different scale of the z-axis in the last column.

Some of the reference results resolve only approximately two waves, while those of Mikkola (2009) give the development of a steady wave train far downstream, and in the first three wave crests there is not much damping. A similar overall trend is observable from the present computations, and those results are then considered the most appropriate for comparison. The computations from the coarse grid give the best correspondence to the first wave lengths such that there is only a tiny phase difference observable. Their amplitudes are however below the other computations. The computations using the fine grid give wave amplitudes that are closer to the results of Mikkola (2009), but the first wave trough is deeper using the fine grid than that of any other computation. The best overall correspondence in both wave height and amplitude in these first waves, but not in the first trough, to the results of Mikkola (2009) and those obtained with the VOF model of Yaffa are obtained from the second grid level of the finer grid. The results from the current tracking model of Finflo show good agreement with the other results in the first wave crest and the two first troughs, but then a phase difference becomes increasingly visible. The results obtained with the VOF model of Yaffa show excellent comparison with those of Mikkola (2009) for the first wave, but then predicts either a small phase difference or deviation in the amplitudes. Those computations give better approximation of the interface than those obtained with the mixture model of Finflo. There is some disturbance left in the first downward slope of the wave, but these are believed to be caused of the far-field boundary conditions, and they posed extended difficulty also in the computations made with the mixture model of Finflo. These will be addressed below. Qualitatively, the formation of the interface and the distributions of the pressure difference correspond well to each other between these eight computations, in addition to the reference results. Experiences from these initial computations, as well as a more quantitative comparison of the all of the results, are given in this section.

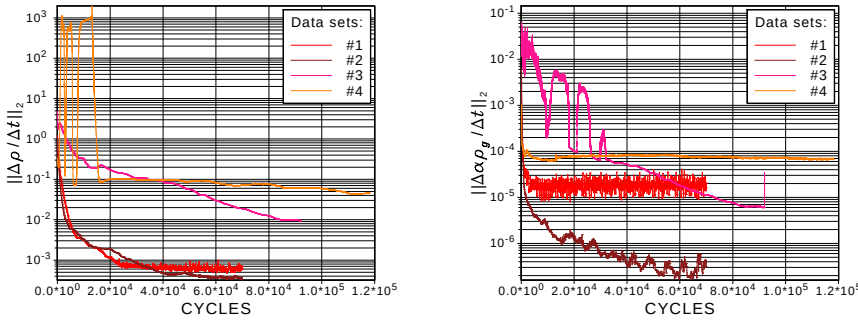
Fig. 4.45 shows the profiles of p_{dif}, p and $|\vec{V}|/V_{in}$ along five different vertical cuts along the x-axis. The topmost row shows those predicted with Yaffa, and the following two rows those obtained with the coarse and fine grids of Finflo, respectively. The presence of the bump at $x = 0$ m is clearly visible in the p_{dif}. It creates almost similar zones of under pressure in the computations on fine grid and with Yaffa, the absolute value being greater from in the computations with Finflo. The other x-locations differ between

these two codes. YAFFA develops peculiar values for the p_{dif} before the bump, which is most likely due to the upstream advancing solitons. FINFLO predicts consistent and almost constant profiles in both phases; however in x-locations before the bump, the presence of the interface is more pronounced than with the reference code, presenting itself as a jump in the profiles of p_{dif}. Both codes give agreeing profiles of p_{dif} for $x = 5$ m, but they differ in sign and magnitude for the last location. This is due to the stronger dampening of the downstream waves. The distributions of the total pressure nearly overlap each other before the bump in computations with FINFLO. They differ somewhat with YAFFA, but also the distributions of p_{dif} were different. Otherwise the distributions of p do not show any significant aspects. There is a wave trough in locations of $x = 0$ m $x = 5$ m which explains the lower levels. The markedly different levels at $x = 10$ m is a result of the same matter; there is an ascending wave slope at $x = 10$ m in the computations with YAFFA and hence the distribution of p is close to that of the previous cut. The notably different shapes from FINFLO at the last cut are explained by the stronger dampening of the waves that far downstream, as well as with the observation that the interface grew in thickness with the vertical distance from the bump. The distributions of the normalized speed show consistent behaviour with YAFFA, but there are some oscillations especially downstream of the bump. The presence of the interface shows itself as a small jolt in the speed. FINFLO gives good predictions for the speed in the liquid phase, but the gas phase shows increased difficulty on both grids. The disturbances before the bump are rather mild. At $x = 0$ m the speed nearly reduces to zero in the gas phase above the interface. This was observed in all computations using FINFLO within this study. The oscillations in this location are more severe using the fine grid. The results on the coarse grid settle to somewhat similar values after the bump when viewed with the finer grid, which show strange behaviour of the gas phase. Throughout the computations there was a peculiar vertical tube of higher speed in the gas phase, and it extended almost the whole length of the domain. This can be a result of the used turbulence model.

Computations on the coarse grid and effects of the Courant number

The parameters that were considered include the discretization scheme chosen for the convection terms as well as the under relaxation factors for the velocity and pressure. The initial studies were performed using the van Albada-limited scheme for the convection of the void fraction. One by one, these could be raised to the final values with the coarse grid to speed up the convergence. The under relaxations for velocity and pressure were set to their usual values, or $\alpha_{u_i} = 0.7$ and $\alpha_p = 0.3$. These were used in all simulations presented on the coarser grid and the van Albada limiter. The order of extrapolation for the velocity was gradually raised to the second-order upwind biased extrapolation in the 12 blocks of interest. The first-order upwind extrapolations were used in the four blocks farthest from the bump (one above and one below the $z = 0$ level on both ends), as well as in the transverse direction since the interest was in the 2D problem. This appeared as a good solution. The convergence was improved since the forming waves, and other disturbances caused by them, remained in a confined zone. This treatment in addition erased another problem, to be dealt with below. The solutions, in terms of the contour $\alpha = 0.5$, on all grid levels of the coarse grid are shown in Fig. 4.43. Those computations were performed without altering the under relaxation factors.

The comparison from computations with different Courant numbers is shown in Fig. 4.46 on the coarse grid. The choice of this seemed to be of great importance. The lowest value fails to obtain a steady state even after 100 000 iterations, whereas those close to the optimum reach this in roughly a third of that computational cost. Additionally, a higher value can show a steady state much earlier, but the level of convergence remains rather modest; however, the total mass balances do go by approximately the same rate using $\mathsf{C} = 5$ and $\mathsf{C} = 10$. This does not tell the whole truth. It is clear from Fig. 4.47 that even while the residuals of the computations using $\mathsf{C} = 5$ advance to the lowest level, the interface resolution in terms of the contour $\alpha = 0.5$ is not improved, but rather it gives a bit clipped approximations to the wave height when compared to the computations using the higher C. The same is observable from computations using $\mathsf{C} = 1$ with even more severe under-prediction, and this time also with an increasing phase difference to the other computations. The computations using the lowest Courant number



(a) Mass balance.

(b) Gas mass balance.

Figure 4.46: Convergence history of the L_2 norm of the mass balance and the gas mass balance with different Courant numbers on the coarse grid. Data set one corresponds to $\mathsf{C} = 10$, two to $\mathsf{C} = 5$, three to $\mathsf{C} = 1$ and four to $\mathsf{C} = 0.1$.

show surprisingly good behaviour in this respect, but it is most likely due to the fact that the interface 'lives on' until the steady state is reached; time-accurate simulations used with the reference code confirm this type of behaviour, as in fact do Figs. 4.43 and 4.51, for instance. In addition, the sharpness of the interface is not improved by using a lower value for the Courant number, and all smear the interface over the same amount of cells; this is further addressed together with the free surface boundary conditions later. Based on these considerations, the $\mathsf{C} = 10$ was chosen for the representative case on the coarse grid with the van Albada-limited discretization for the void fraction.

The iteration history of the contour $\alpha = 0.5$ is plotted in Fig. 4.48. It shows that the final shape is obtained very early in the solution. This is due to the fact that a steady level was reached with all grid levels. The markedly quick settlement to this steady value of the computations using the $\mathsf{C} = 10$ was then investigated. Judging by Fig. 4.48, the contour of $\alpha = 0.5$ reaches its steady value within the order of mere thousands of iterations. This was an unexpected turn of events within this study, despite the fact that the convergence of it did settle, but the other residuals followed upon that of the mass balance. Further analyses shown in the same figure indeed indicate that it remained very closely the same throughout the computations. A closer

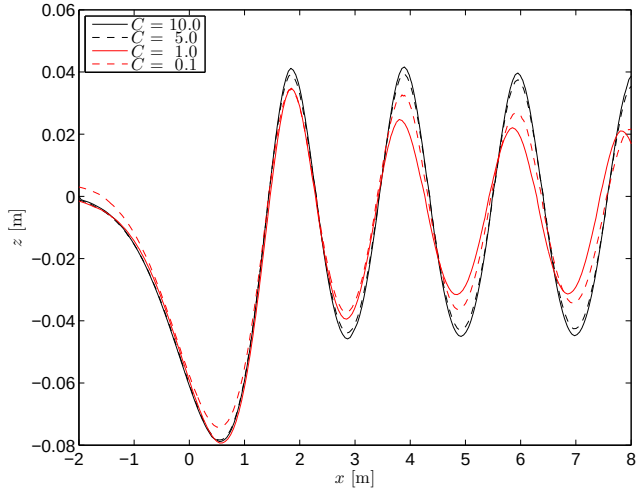
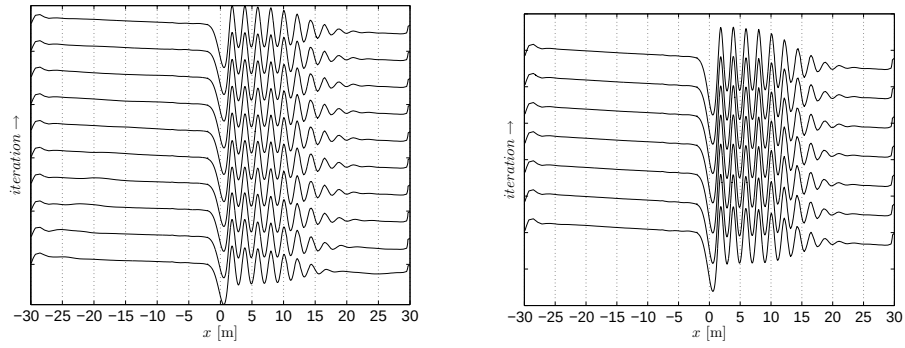


Figure 4.47: Comparison of the interface profiles from computations with different Courant numbers on the coarser grid. The lines correspond to contours of $\alpha = 0.5$.

look at the first wave crest, shown in Fig. 4.50, reveals that the amplitude does change but only on the order of the cell size. However, it is evident in the plots that there exists a rather interesting downward slope towards the downstream boundary of the domain throughout the computations. A possible cause of this is discussed later.

Computations on the fine grid

The computations done on the fine grid reveal more serious problems of the study. The finer grid was over four times denser in the x-direction than the coarse one. This required significantly more under-relaxation in the algorithm, and it was found that using as low a value as $\mathsf{C} = 0.1$ was the only option. Due to the larger amount of cells, and to a non-standing access to 16 processors, the computations were very time consuming. For example, on the third grid level the calculations were run for 250 000 cycles, and still the residuals maintained their steady descend. The convergence curves from the three grid levels are shown in Fig. 4.49. The jumps that can be seen in all grid levels on the gas mass balance were typical for situations where there was some instability observable in the α distribution but it seems to stabilize



(a) One contour denotes every 1000th iteration.

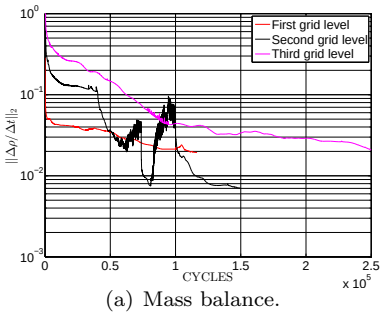
(b) One contour denotes every 10000th iteration.

Figure 4.48: Iteration history of the shape of the contour $\alpha = 0.5$ on the coarser grid.

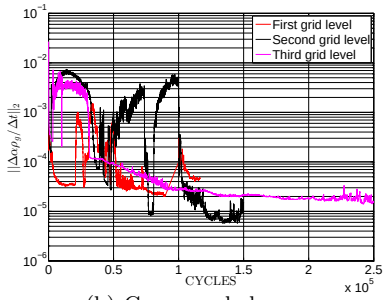
as the iteration is carried on. The use of a larger Courant number ultimately crashed the computations. The most usual case was a continuation of the instabilities and a bizarre break-up of the interface at some part along the first waves, leading to the crash of the solver. In those cases the wave seemed as if it started to break, or to turn inwards, even at such mild slopes. Naturally, this by itself should not crash the solver. Perhaps the dispersion-type errors were not smeared but propagated still along the fine grid. The under-relaxation factors for the velocity and pressure did not prove to be as important a measure. However, it was not possible to extend the study in the previous degree to these values. Some fine tuning was earlier allocated to the MUSCL formula in each index direction. The similar damping zones were used here as well, but the solution was stabilized using the second-order upwind method for the interesting blocks. The results from the second grid level give better predictions for few downstream waves, and the solution is plotted also in Fig. 4.44.

The boundary conditions

The denser blocks of the finer grid were located symmetrically around the bump. The grid spacing in the x-direction was significantly smaller, and hence enabled the disturbances to propagate easier in both negative and positive directions, even on the third grid level. This is notable in Fig.



(a) Mass balance.



(b) Gas mass balance.

Figure 4.49: Convergence history of the L_2 norm of the mass balance and the gas mass balance with different Courant numbers on the fine grid.

4.51, where some longer waves are transmitted upstream of the bump even though the simulations were not accurate in time. It seems that not even the first-order extrapolation, employed for the velocity in all index directions, was enough to erase these. Additionally, the 'damping zone' is clearly too fine, even on third grid levels. It is here noted that when the first-order extrapolation was tried in all blocks of the finer grid, also the downstream wave train damped completely after the first wave. Obviously the term Δx was small enough not to very notably affect the upstream waves in the former case, though the waves do decrease in amplitude as they approach the inflow boundary. The grid spacing in the z-direction around the interface stays the same throughout the domain. The same phenomenon was observed by Lowery and Liapis (1999), although they studied a submerged semicircle. Mikkola (1999) reports similar findings. On the coarse grid, the shape of the interface does not significantly change after a mere 1000 iteration cycles, and the possible upstream solitons are dampened very early in the iteration. This marvellous achievement is mainly due to the well converged results from the previous grid levels. In Fig. 4.43 there is next to no visible phase difference between the grid levels, for instance, and Fig. 4.52 shows the change of the average wave length to be of the order of the one cell width on the first grid level. On the fine grid, on the other hand, such is still observable, most notably from the third grid level (even at such high amount of cycles). Note that also in Fig. 4.52 all slopes of the coarser grid are linear.

Computations on the fine grid presents one great challenge in these types of computations. As was mentioned earlier, problems arise from not having

sufficient information of the flow solution outside the domain boundaries, and the approximations made to comply this. The simplification made here was the extension of the domain towards infinity, that is, assuming initial undisturbed states at the inflow and outflow boundaries. This corresponds to the numerical beach approach in the literature. The approximation of still inlet and outlet conditions would require that all disturbances created inside the domain vanish at these boundaries. Fig. 4.51 shows that this is not the case; even the first-order discretization for the velocity did not suffice. The free surface exhibits unsteady behaviour at both the inflow and outflow boundaries. The upstream advancing solitons seem to give rise to reflections in the inflow boundary. After 60 000 cycles, the water level also near the outflow boundary begins to climb, and seems to shed a very long wave to the solution domain. From the stationary solutions obtained from the coarse grid the contour of $\alpha = 0.5$ settles to a mild downward slope in the direction of the flow (it might seem as if there was one very long wave) instead of the level of $z = 0$, accompanied with the jumps at the inflow and outflow for the values to correspond to those of the undisturbed situation.

The casting of the boundary conditions is a very difficult topic especially in cases where there are supposed to be waves in pressure and velocity in the domain, and their extent is not known *a priori*. This necessitates future studies on these far-field boundary conditions. Quite self-explanatory the grids used here were too fine for the adequate numerical diffusion to take place, even on the third grid levels. But instead they, rather impishly, served in providing the mentioned illumination. Time-accurate computations done with the reference code showed very clearly the problem created by these reflecting waves. It is most likely those that presented a big problem in the calculations, and a reminiscent situation is clearly seen in Fig. 4.44 in the results of the code YAFFA. These did not appear to diminish in time, but continued their propagation downstream as well as upstream directions within the domain. This is very unfortunate, and one main reason for presentation of the results on the rather coarse grid; in this sense they merit the consideration of ad hoc information. The discussion is resumed in Chap. 5. A mirror boundary condition was used in other parts of the domain, except for the slip or no-slip wall on the bottom, and no problems were encountered from these.

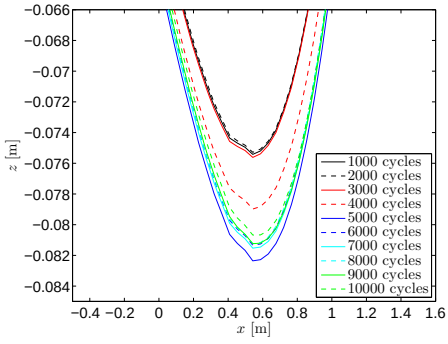


Figure 4.50: Iteration history of the shape of the first wave crest in terms of the contours $\alpha = 0.5$ on the coarser grid on the first grid level.

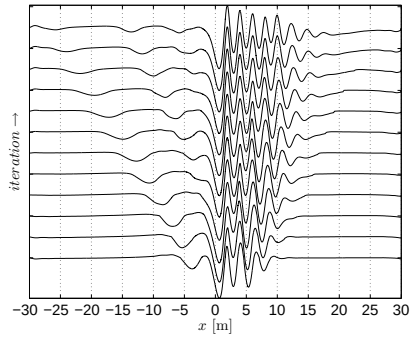


Figure 4.51: Iteration history of the shape of the contour $\alpha = 0.5$ on the finer grid. One contour denotes every 10000th iteration on the first grid level.

Convergence upon grid refinement

The computations were carried out with FINFLO with six different grids. Fig. 4.43 readily provided the interface shapes obtained from all of these. The grid convergence is briefly considered here by simply doubling the amount of cells in every index direction, and the six different grids correspond to the three grid levels on the coarser and finer grids. The convergence is checked with the wave lengths, amplitudes and heights, and these are given for all cases in Tab. 4.4, and in Fig. 4.52. The computations with the current tracking model of FINFLO show greater wave heights than any other computation by over 10 cm, but the wave amplitudes and heights are all within 3 cm. An exception to the wave amplitudes and heights follows from the computations on the third level of the coarse grid, since the resolution is poor. The discrepancy in the wave lengths is greater than in the wave amplitudes, but the absolute values of the wave lengths are roughly fifteen times larger than the wave heights.

From Fig. 4.43 it can be seen that the coarser grid shows uniform convergence upon grid refinement. The interface shape becomes sharper as the grid is refined. As could be expected, and there is no pronounced phase difference between the waves. This is due to the fact that a steady-state result was accomplished in all grid levels, and can be seen as the linear plots in Fig. 4.52. The differences in the wave lengths are of the order of the grid

Table 4.4: Observed wave lengths λ, wave amplitudes A and wave heights h from the computations. The number of waves n denotes that used in the average.

Computation	λ [m]	n	A_{max} [m]	A_{min} [m]	h_{max} [m]
YAFFA VOF	2.06	7	0.052	−0.079	0.141
FINFLO, coarse grid, 1st level	2.12	8	0.041	−0.079	0.120
FINFLO, coarse grid, 2st level	2.09	6	0.031	−0.074	0.105
FINFLO, coarse grid, 3st level	2.07	3	0.021	−0.070	0.091
FINFLO, fine grid, 1st level	2.12	6	0.054	−0.096	0.150
FINFLO, fine grid, 2st level	2.11	5	0.054	−0.077	0.131
FINFLO, fine grid, 3st level	1.98	4	0.046	−0.074	0.120
FINFLO, moving grid	2.24	7	0.054	−0.079	0.133
Mikkola (2009)	2.07	3	0.062	−0.088	0.150
Mikkola (1999)	2.11	2	0.050	−0.088	0.138
Bet et al. (1996)	2.11	2	0.050	−0.088	0.138

spacing.

From Fig. 4.43 it is evident that a no steady solution was obtained on the finer grids. Some discussion was presented earlier on the possible causes of this. A clear phase difference is observable between the solutions on the third grid level and those on the two finer grid levels. The convergence of the interface shape, or the contour $\alpha = 0.5$, does not evolve as expected, and the second grid level provides better estimations for most waves than the finer grid does. Reasons for this were readily presented. These issues manifest themselves as the non-linear convergence curves in Fig. 4.52.

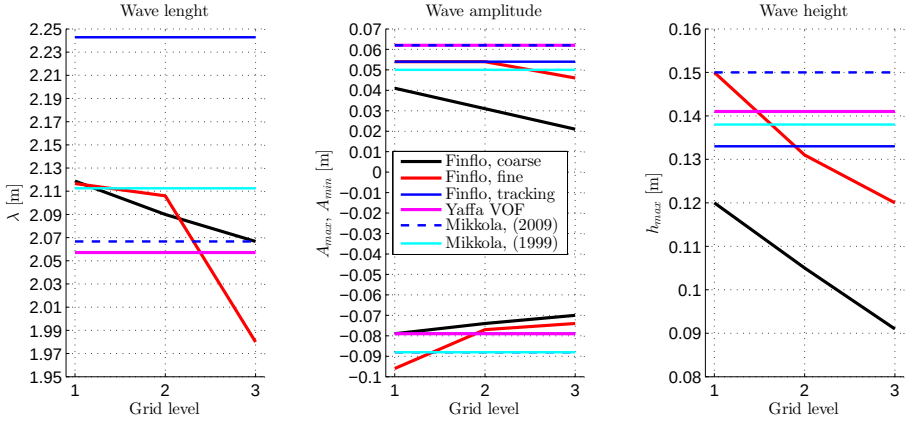


Figure 4.52: Convergence of the wave length, amplitude and height upon grid refinement. Note that the results of Mikkola (1999) and Bet et al. (1996) are essentially the same.

The discretization scheme for the convection of the void fraction

The bump case was computed using two different limiter functions for the MUSCL extrapolation. These were the van Albada and SUPERBEE limiters. They were introduced in Sec. 3.3.2 and studied more thoroughly in the previous section. Here the purpose is to study their performance with the Navier-Stokes solver. Once the solution had reached a steady level on the first grid level using the van Albada limiter, the computations were continued with the SUPERBEE-limited flux calculation for the α equation.

The continued computations proved very unstable. The compressive limiter required more under-relaxation in the algorithm. Now the relaxation factors for velocity and pressure appeared more influential, and when lowered to $\alpha_{u_i} = 0.25$ and $\alpha_p = 0.01$ the computations stabilized on both grids. For the same reason, the Courant number had to be lowered to $\mathsf{C} = 0.1$ on the coarse grid and to $\mathsf{C} = 0.07$ on the fine grid. These consequently slowed the convergence down so significantly, that a steady-state solution was not possible to be reached within this work. The cycles amount to around 150 000, on a single case, on both grids and on the first grid levels.

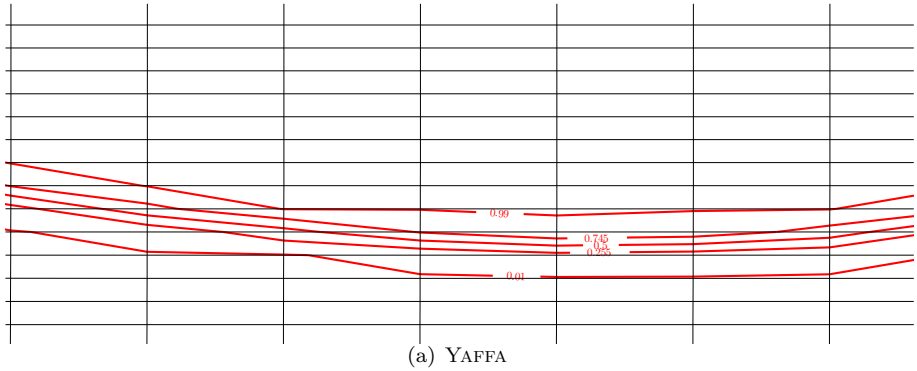
A comparison in terms of the contour $\alpha = 0.5$ using the different limiters and grids is shown in Fig. 4.43. The plots show very little difference in the free surface resolutions, and the two contours mainly overlap each other. On the fine grid, the SUPERBEE seems to present the water side slightly lower than the van Albada limiter, except for the beginning of the first trough. On the coarse grid, no conclusion can be drawn based on the contour.

A closer inspection of the resolution in the z-direction using the both limiters is shown in Fig. 4.53, together with the computational grids, as well as the corresponding plot from YAFFA. The figure shows the second wave trough and the contours of $\alpha = [0.010, 0.255, 0.500, 0.745, 0.990]$. Based on these limits, the interface is smeared over $\mathcal{O}(35)$ ($\approx$ 12 cm) cells on the coarse grid using the van Albada limiter, and over $\mathcal{O}(25)$ ($\approx$ 8 cm) using the SUPERBEE limiter. On the fine grid, the interface is smeared over $\mathcal{O}(25)$ ($\approx$ 8 cm) cells using the van Albada limiter, and now the SUPERBEE limited is sharper by only two cells on the water side. The contours of $\alpha = 0.5$ nearly overlap each other in both cases (cf. Fig. 4.43). As could be expected, the IGDS approximates the interface within two to three cells, analogously to

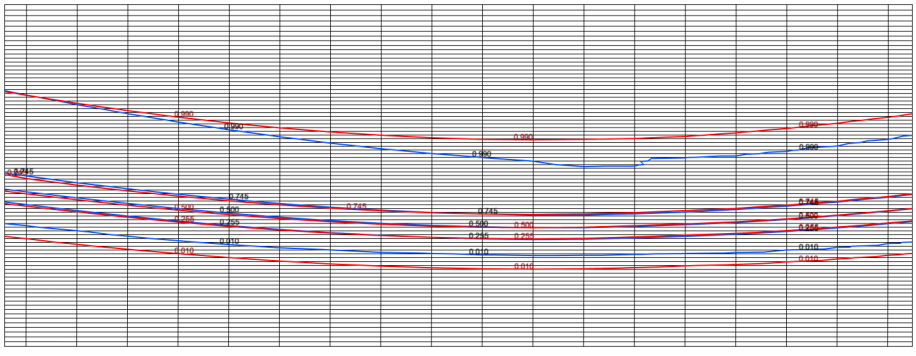
the previous section. Also its highly compressive behaviour is clearly visible in this computation as well; this could have caused some of the oscillations that are still persistent. The SUPERBEE limiter performs somewhat as expected on the coarse grid, and provides some compression to the steady-state interface obtained with the van Albada limiter. As the grid is refined in the x-direction, also the van Albada-limited predictions are sharper, and also slightly more favourable than the SUPERBEE-limited. This is a bit confusing based on the results from the previous section, but FINFLO uses the Euler implicit method in the steady-state solution. Additionally, the computations made on the first level of the finer grid failed to moderate. However, the SUPERBEE limiter gave slightly better predictions for the fulfilment of the free surface boundary conditions, and these will be discussed next.

The free surface boundary conditions

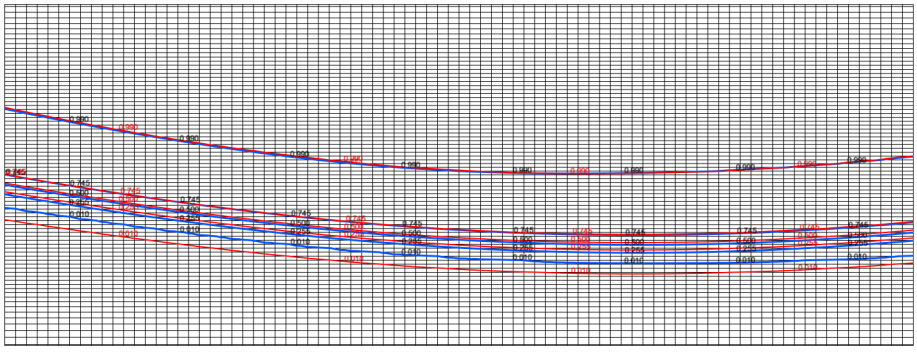
The free surface boundary conditions corresponding to the kinematic one is related to the solution of the volume fraction equation. The continuity equation reduced to the volume fraction equation, and also the satisfaction of the mass balance in the iterative solution, *i.e.* the pressure correction stage, reflects into this. Fig. 4.46 shows the convergence of the mass balance and the gas mass balance for the first level of the coarse grid. It is clear that the machine accuracy is not reached but, whilst dropping several orders of magnitude, the residuals remain oscillating around an apparently steady level. The gas mass balance shows better and faster convergence than the mixture mass balance. Although the volume fraction residuals converge to a lower level using the $\mathsf{C} = 1$ and $\mathsf{C} = 5$, as was mentioned earlier, the solution obtained with $\mathsf{C} = 10$ gave interface shapes that were in a better agreement with the reference solutions. This convergence can however influence the rather smeared interface observed as well, and further study is needed based on the experiences learned from the previous section. It is noted, however, that the estimations for the interface thickness were not improved when using $\mathsf{C} = 5$, for instance. Convergence histories from the finer grid are shown in Fig. 4.49. The gas mass balance reaches comparative levels, but the mixture mass balance is still over an order of magnitude higher when viewed with the coarse grid.



(a) YAFFA



(b) Coarse grid



(c) Fine grid

Figure 4.53: Comparison of the resolution of the interface from computations with YAFFA and with the two limiters. In the results with FINFLO, the red lines denote computations using the van Albada limiter and the blue lines those with the SUPERBEE limiter. The lines correspond to contours of $\alpha = [0.010, 0.255, 0.500, 0.745, 0.990]$. This figure shows the second wave trough together with the computational grid. The horizontal coordinate ranges as $x = [2.4, 3.2]$ m and the vertical as $z = [-0.15, 0.15]$ m.

The dynamic boundary condition is related to the total pressure in the vicinity of the free surface. If the interface is seen as a transition zone with zero thickness, the pressure on it equals to the atmospheric pressure at this level. Here, that is investigated the following way. The contours of the void fraction $\alpha = [0.1, 0.5, 0.9]$ are taken to represent the computed interface, and its finite thickness. These are plotted together with the distributions of the total pressure along several wave locations, and shown in Apps. B.1 and B.2 for computations from the coarse and fine grids, respectively, and in App. B.3 for computations from YAFFA. The results from FINFLO are given from both the van Albada and SUPERBEE-limited calculations, and the constant ambient pressure p_∞ is subtracted from the total pressure for the post-processing. The inspections are made from the crests and troughs of the first and third waves.

The first wave crest shows a peculiar behaviour of the interface from both grids of the FINFLO computations. The total pressure seems to spread up to the gas phase, and this tendency increases on the finer grid. The SUPERBEE limiting seems to compress this mildly, but still the total pressure smears far to the gas phase. The pressure from the liquid side however remains sharp, but that is due to the hydrostatic part. The same behaviour is observable on the first wave crest, but this time slightly milder. Again, the SUPERBEE limiting provides more compression. The results from the third wave trough and crest then again shows much more confined zone for the total pressure. The use of the denser grid does not seem to have a pronounced effect in the sharpness of the pressure distribution. Both grids and limiters show sharper approximations to the transition zone as well as the pressure in the wave troughs than in the wave crests. The SUPERBEE-limited computations do not show much improvement on the fine grid.

The computations with YAFFA were done with the very compressive IGDS discretization for the void fraction equation, as well as with the modifications to the bulk flow solver to account for the contact discontinuity. These features together with the coarser grid employed still provide superb free surface resolution in terms of the dynamic boundary condition. The pressure jump is always confined in the narrow zone between the volume fraction values of $\alpha_g = [0.5, 0.9]$. The first wave is as sharp as the third one. The seemingly unfortunate smearing of the interface is due to the rather coarse grid that was used.

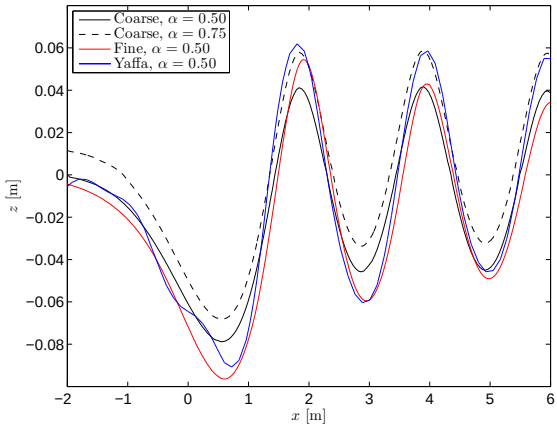


Figure 4.54: Comparison of the interface profiles from computations with different definition of the contour on the coarser grid. Computations from the fine grid and with YAFFA are also shown using the traditional definition.

At best the mixture model of FINFLO captures the dynamic boundary condition *almost* within the zone $\alpha_g = [0.5, 0.9]$ as well. The SUPERBEE-limited computations manage to squeeze all but the last $\mathcal{O}(20)$ Pa in this zone. A very mildly increased error is made with the van Albada. It also seems that in terms of this boundary condition, the contour of $\alpha = 0.5$ does not represent the free surface, but rather some value in the aforementioned zone. The same is observable from the results from YAFFA, but prior to a more decisive comment, a more thorough study should be conducted using that code. The results from the coarse grid, in terms of the contour $\alpha = 0.75$, are plotted in Fig. 4.54. It reveals that this treatment does not produce any improvement in the prediction of the interface, but merely lifts it up as the interface is nearly equally thick throughout. It can be interpreted that the interface is located somewhere near this zone, most likely toward the traditional contour; further grid refinement studies are needed. Finally, within this work, it can be concluded that the mixture model of FINFLO makes an error of $\mathcal{O}(50)$ Pa in capturing the dynamic boundary condition on the free surface.

Chapter 5

Closure

The intention of this work was to verify the applicability of the homogeneous mixture model for the free surface problem. In a sense this could be seen as rather straightforward, since the problems involving the free surface, as described in Sec. 1.2, are a special case in multiphase flows, and the model used here is a more general approach in that category. The special nature mainly stems from the fact that the immiscible phases are segregated. This was described in detail in Sec. 1.1. It is important to understand that this speciality does not automatically guarantee the applicability of the more general approach to the precise problem at hand. It should be clear by now that a computer program that numerically solves the Navier-Stokes equations for practical flow cases, is a complex composition of a group of numerical methods that are specifically designed for such problems. The successful utilization of such a composition to other complex flow cases is not by any means obvious. The inspection of the numerical methods was one attempt to understand the aforementioned invocation. The betaken quest is then in essence the verification of the model, and this will be concluded below.

Verification of the mixture model

The process of verification can be stated following Oberkampf and Trucano (2002): "Effectively, verification provides evidence (substantiation) that the conceptual (continuum mathematics) model is solved correctly by the discrete mathematics computer code. Verification does not address whether the conceptual model has any relationship to the real world." It is appropriate to note that the model is verified *for* the problem; the formulation of the problem was mathematical in the respect that no stand was given for the relationship to the real world *per sé*. Here, one asserts both that *the* right equations are solved (Chap. 1 and Chap. 2), and that they are solved in *a* right way (Chap. 3 and Chap. 4).

The title concerning the verification of the model is clarified and summarized in the following. It was discussed in Sec. 1.1 that the mixture model is indeed a direct consequence of the conditional averaging procedure which has been exercised to obtain such a (numerical) model that can distinguish between two segregated phases. Such system is represented by water and air, for instance. Therefore, it is deducible that the foundations upon which the model is built do lay in the region that governs the free surface problem, and hence the notion of a special case is rather obvious. One however needs to obtain that verification prior to proceeding to the actual validation of the model, and that verification is composed of several parts.

The use of the volume fraction transport equation determines the shape of the interface (free surface) between the two fluids in traditional interface capturing models that use the VOF strategy. Additionally, it can be seen as the kinematic free surface boundary condition. The present model belongs to this category, but the way in which this equation is solved differs from the traditional one. As was noticed, this is a rather distinctive feature of the algorithms used in FINFLO. Thus far it is concluded that the methodology for the capturing of the free surface is legitimate, and this argument is based on the discussion presented in Sec. 4.2.

The approximation of the same averaged velocity and pressure fields for both phases is the unified approach in the VOF strategy. That is, the equations are formally the same as in the case of single phase flow, and a discrimination of one specific field can be made from the indicator. Again, if

one craves a more accurate depiction of the interface of the separated phases, there are myriad of modelling possibilities. It is seen, however, that these types of approaches would need justification of the sort that one interests in the true interface phenomena, whereas the practice is instead to focus on more appropriate scales. The present motivation has additionally always originated from a perspective of a naval architect. It can then be concluded that the use of such conditioned (and averaged) equations is, thus far, appropriate for the free surface problem, as the problem was defined in this thesis. This justifies the simplifications made earlier, and adds to the verification of the model.

Ideally, the interface would exist only between one or two cells. Only then, strictly speaking, the assumptions imposed in Sec. 1.1 would be valid in a converged approximative solution. And only then, it is possible to satisfy both of the free surface boundary conditions. Requirements for the algorithm to achieve this were already discussed. But if we solve the pressure differences instead of the pressure, as was the juxtaposition here, the momentum equations would need reformulation due to the change of the working variables. This is discussed later. It is hypothesized that the error made earlier could then be diminished. Additionally the continuity equations were under-relaxed to account for quite different physics. This is addressed below as well. Be that as it may, one cannot help but to cogitate if, perhaps, there is a trade-off between the strong solution of the governing equations and the fulfilment of the boundary conditions.

Next, attention is given to individual phases of the solution process. This is done in order to establish the verification, as it was given earlier and in Sec. 1.4, as well as to pinpoint and give remarks on issues that need revision before proceeding to the validation which is obviously the next priority.

The solution algorithm

The solution algorithms in the free surface capturing type strategies commonly reduce to those of the single continuum mixture. Such a model can be constructed by adding a transport equation for the indicator to the underlying single-phase flow model. Conversely, such a model can be attained by reducing an existing multi-phase model to comply with the previous require-

ments only. As was seen, the application of such model as it is can produce good results also for these external problems. But now the model of FINFLO was designed for cavitating and boiling flows, that is, in cases where phase change, and consequently mass transfer, occur. The implementations and fine tunings were done focusing on the extension of the code into these very demanding phenomena. This is to say the transition to the free surface capturing VOF model is not ready, but promising results are available.

There are numerous ways to construct a SIMPLE algorithm. Two variants were included in this study. In the mixture model of FINFLO, the explicit stage is the same for both the pressure- and density-based algorithms. Additionally some ideas originating from the density-based methods had been incorporated in the present model. The algorithms of FINFLO for the solution of the indicator are based on the mass balance, whereas other implementations prefer the use of volumetric balances also in the implicit stage. Then the mass balance is indirectly enforced with the solution of the volume fraction equation (cf. Sec. 2.2), and the pressure correction stage acts to account for the pressure, and conservative volumetric fluxes. At present this deviation had been done also because of the close resemblance to the solution of the momentum equations. Then again, if purely the primitive variables are solved, which is possible since the phases are assumed incompressible, also the momentum equations readily reduce to the same form. In principle, then the momentum fluxes cannot be discretized in a conservative way[1]. Another novel feature of FINFLO in the SIMPLE strategies was the transformation of the explicit residuals, which commonly is done only in the density-based methods. The algorithms of FINFLO for the solution of α present thus a novel approach to the problem.

The flux calculation

The calculation of the momentum flux can be done similarly as in the case of single phase flow, since the interface is in fact a streamline in steady-state situations. Traditionally, the flux-difference splitting of Roe has been used. A problem arises in the case of the void fraction equation, and this has been discussed extensively. In transient computations, the conservative flux of

[1]It could be hypothesized that the thicker the interface, the more severe conservation error is made in the solution.

momentum comes also with a discontinuity, but the Roe scheme should be able to tolerate this when a suitable scheme is used for α.

In this work, the viscous fluxes do not provide intrinsic difficulty since the no-slip condition was applied in regions of essentially unit phase fraction. Then, the material properties are constant (the homogeneous mixture approximation provides for the inter-phasic slip). The coefficient of viscosity is also a changing material property, and consideration should be put on its behaviour in the discretized equations. Additionally, the nature of turbulence in the presence of a mixture is not a well-known subject, and rather it is assumed that, for example, the viscosity varies linearly through the interface. In practice, the standard models are used, whereas in ship flow cases, it is the location of the interface on the ship hull that can bear a great influence.

The pressure correction stage

It is interesting that the pressure correction stage of Finflo is derived from the mass balances, whereas that in Yaffa follows from the incompressibility condition. In fact, the latter is the approach that is commonly taken in these types of flow problems, *e.g.* by Darwish and Moukalled (2006), Moukalled and Darwish (2012), Vaz et al. (2009), Queutey and Visonneau (2007), Ubbink (1997) and Kissling et al. (2010). This stems from the fact that the fluid densities can differ significantly, and Rusche (2002) notes that the lighter phase hardly influences the mixture mass imbalance.

The pressure correction equations were written for both variants earlier in this thesis. It was shown that by making certain simplifications, the phasic continuity equations reduce to the form where the driving force becomes the mixture mass imbalance, or sums of the phasic mass balance errors, but weighed with the component densities. The equation then eludes the previous problem, and is suitable for these types of computations. Rusche (2002) and Queutey and Visonneau (2007) formulate the semi-discretized momentum equation, resulting to the pressure corrections, differently but both retain the gravity term accompanying the pressure gradient in the

intermediate step

$$\vec{u}_P = \frac{\sum_{nb} A_{nb}\vec{u}_{nb} + R_P}{A_P} - \frac{\nabla p}{A_P} - \frac{\overrightarrow{\mathscr{G}}}{A_P} \ , \tag{5.1}$$

where the coefficients A_P and A_{nb} are the diagonal and off-diagonal terms resulting from the linearizations of the momentum equations, respectively, $\vec{u}_{nb}$ the velocities in these cells neighbouring the cell P, known after the solution of the momentum equations, R_P the explicit source terms (without the gravity and pressure), and $\overrightarrow{\mathscr{G}}$ the contribution from the gravity. This velocity is then used to enforce the continuity constraint. Furthermore, Queutey and Visonneau (2007) present a novel approach to account for the pressure gradient discontinuity in the SIMPLE algorithm, by instead basing the pressure correction equation on $\nabla p/\rho$ which is continuous across the interface; then the gravity term reduces to $\overrightarrow{\mathscr{G}} = \vec{g}$. A similar formulation[2] was used in the construction of the pressure correction equation of YAFFA in Sec. 3.2.5. If the solution is sought using the conservative variables, the novelty is then based in dividing the semi-discretized momentum equations by the cell centre density, and extrapolation of that velocity to the cell faces to form the volumetric fluxes. That is, the pressure gradient based on $\hat{p}$ is used in Eq. (5.1); consequently the equation is then multiplied by an expression involving the same ρ_P as noted previously. When that velocity is extrapolated to the cell faces, the interpolation of the densities in the continuous gradient is based on Eq. (3.86). This could be *interpreted* in the present notation through the momentum equation which is constructed similarly as in Sec. 3.2.5

$$A_l u'_{i,l} + \sum_m a_{lm} u'_{i,m} = -\sum_{lm} S_{lm} p'_{lm} n_{i,lm} \ , \tag{5.2}$$

where i denotes a vector component; the off-diagonal coefficients are now $a_{lm} = -(\rho S)_{lm}\mathsf{max}(-\bar{u}_{lm}, 0)$ and the diagonal one $A_l = \rho_l \mathcal{V}_l/\Delta t - \sum_m a_{lm}$. The division by the density of the cell l leads to

$$\left(\frac{\mathcal{V}_l}{\Delta t} - \sum_m \frac{a_{lm}}{\rho_l}\right) u'_{i,l} + \sum_m \frac{a_{lm}}{\rho_l} u'_{i,m} = -\sum_{lm} S_{lm} \frac{p'_{lm}}{\rho_l} n_{i,lm} \ . \tag{5.3}$$

[2]The authors discuss that the gravity term must be retained with the pressure gradient in the pressure correction equation so that the pure hydrostatic equilibrium is satisfied. This had not been done with YAFFA.

If one uses the SIMPLE approach, the velocity correction results in

$$u'_{i,l} = -\left(\frac{\mathcal{V}_l}{\Delta t} - \sum_m \frac{a_{lm}}{\rho_l}\right)^{-1} \cdot \sum_{lm} S_{lm} \frac{p'_{lm}}{\rho_l} n_{i,lm} \ . \tag{5.4}$$

The correction of the convection velocity, which is used in the equation for the volumetric balance, is obtained by taking a dot product of Eq. (5.4) and the cell face normal $n_{i,lm}$, and using suitable averages for these convection velocities at the cell centres, the coefficients and geometric terms. If the cell face pressures are transformed into gradients with the Gauss theorem, this form in fact reduces to the same that was used earlier, but with the exception that the densities were then based on more local information (cf. Sec. 3.2.2). Despite the success, these ideas might be worth considering with FINFLO as well, and this could be achieved by a similar manipulation for the coefficient A_P that result from the linearizations of the momentum equations, Eq. (3.30). At the moment it is not clear, however, whether this is necessary since the working variables were different. It is here noted that the interpolation of the coefficient A_l to the cell faces posed a great problem in the development process[3] of the code YAFFA when the jumps in the density were present, and these were solved by removing the density.

With the non-constant density in the domain, the gradient of the modified pressure was $\nabla p = \nabla p_{dif} + \rho\vec{g} + \vec{g} \cdot \vec{x} \ \nabla\rho$. Insertion of this into the momentum equations gives

$$\frac{\partial \rho \vec{V}}{\partial t} + \nabla \cdot \rho \vec{V} \vec{V} = -\nabla p_{dif} + \nabla \cdot \tau_{ij} - \vec{g} \cdot \vec{x} \ \nabla\rho \ . \tag{5.5}$$

The difference compared to the conventional form of momentum equation is that the density gradient now appears near the interface, and the other gravity source vanishes. Rusche (2002) notes that this form enables efficient numerical treatment of the steep density jump at the interface, by including the term $\vec{g} \cdot \vec{x} \ \nabla\rho$ to the Rhie-Chow interpolation. Now the gravity contribution in Eq. (5.1) is $\overrightarrow{\mathscr{G}} = \vec{g} \cdot \vec{x} \ \nabla\rho$. The density gradient could be approximated with central differences, or transferring it to the surface integral via the Gauss theorem. This is a rather new approach in a single pressure system, and it has not been detailed by anyone but Rusche (2002). A natural interpretation to relate this to the present work is simply the reformulation

[3]Private communications with D.Sc. Satu Hänninen.

of the pressure correction stage, described in Sec. 3.2.5, such that this term is retained. This notion applies only to the solution procedure of FINFLO, of course.

In this work, it was assumed that the volume fractions of the two-phases can be combined, *i.e.*, the other phase is always obtainable from $\alpha_1 = 1-\alpha_2$. If this is true also in the numerical solution, the velocity field must be strictly divergence-free in incompressible cases; cf. Sec. 2.3, Lafaurie et al. (1994) and Ferziger and Perić (1999). This is easily seen via the substitution to Eq. (3.99) for a computational cell i

$$\mathcal{V}_i \frac{\Delta(1-\alpha_{i,2})}{\Delta t} = -\sum_{faces} \Big(\bar{u}(1-\alpha_2)S\Big) \tag{5.6}$$

which yields

$$\mathcal{V}_i \frac{\Delta\alpha_{i,2}}{\Delta t} = -\sum_{faces} \Big(\bar{u}\alpha_2 S\Big) + \overbrace{\sum_{faces} \Big(\bar{u}S\Big)}^{\hat{=}\nabla\cdot\vec{V}_i} \ . \tag{5.7}$$

As was discussed a few times before, the enforcement of this condition is mainly done via the pressure correction stage. However, the novelty of the solution algorithms of FINFLO is based also on the subtraction of the error in mass balance from the conservative residuals, when they are transformed to those of the primitive variables; see *e.g.* Eq. (3.92). The pressure correction stage was discussed previously, but this in conjunction with the solution for α (or x in case of the mass balance) would be an interesting topic for future research. For example, one could blend the volumetric balance and that of the auxiliary phase in the pressure-correction equation, and thus provide better coupling between α and p.

The indicator equation

The motivation for the use of the gas mass fractions came partly from the possibility to use unified expressions for the implicit stage. The mass fraction is however considerably smaller than the volume fraction, and *e.g.* the value of $\alpha = 0.5$ corresponds to $x_g \approx 0.001$, and $x_g = 0.5$ to $\alpha \approx 0.999$. If the mass fractions were extrapolation to the cell faces in the explicit stage, this could diminish accuracy. The relationship between x_g and α is plotted in

Fig. 5.1. The value of the void fraction goes through a steep change as the mass fraction approaches the value 1. In the vicinity of the interface, if one defines it as the traditional iso-surface, the slope of the mass fraction is very small. That is, tiny changes in x_g ($\mathcal{O}(10^{-3})$) can produce even two orders of magnitude larger variations in the void fraction. If one desires this to be confined, problems could arise in the numerical algorithm considering these differences since the mass fluxes can vary considerably. It could seem to be advisable to base the solution process, at least for the interface evolution, on the α equation alone. The mass fraction, however, goes through a very sharp transition near the lighter phase boundary; a closer study of these aspects would be intriguing.

Sipilä (2012) used the homogeneous mixture model of FINFLO, sometimes referred to as the cavitation model, in RANS analyses of cavitating propeller flows. He, contrary to the present work, reports capturing half of the cavitating propeller tip vortex[4] within nine computational cells. Here, the interface was captured with over 20 cells at best. In the original model, no sharpening of the α distribution was sought. These discrepancies are rather unexpected, but it must be remembered that the equations that are solved were not the same. In the cavitation model, the right hand side of the void fraction equation is non-zero, instead consisting of a mass transfer model, since the equation is the continuity equation for the gas phase. The calculation of the explicit residuals is different since the source term is added to the differences in the fluxes. Additionally, the source term can be linearized for the implicit stage to provide more diagonal dominance than the pseudo-time derivative alone. Out of these differences, it is most likely the former that provides sharper predictions, whereas the latter could have a greater influence on the robustness. Sipilä (2012) performed systematic tests varying the coefficients of the mass transfer model, and studied their effect on the cavitating tip vortex. It was found from those results that a tad more 'compression', in present terms, was achievable. This depended on the coefficients of the evaporation and condensation used in the model, whereas some other values for those coefficients came with more smearing in this sense. In the VOF models, the continuity equations do not have source terms, and the only possibility to affect the sharpness in that equation in principle comes from the treatment of the convection. Some observations on the robustness were readily made in the previous chapter.

[4]The cavitating tip vortex core is filled with almost pure vapour.

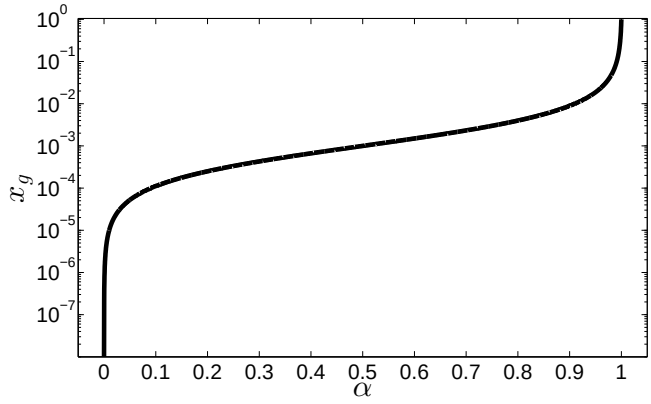


Figure 5.1: Gas mass fraction as a function of the gas volume fraction.

The results of the convection problem presented in Sec. 4.1 demonstrated the outstanding performance of the SUPERBEE-limited MUSCL differencing scheme together with a few schemes based on the NVF. A similar correspondence was not seen in the results of the bump case, merely a favourable trend. Again, this was not expected, but few plausible reasons can be pointed out. As was mentioned in Sec. 4.2, there were significant convergence problems when the more compressive limiter was turned on. No computations using this could be continued so that a steady state was reached within this work. These are discussed more below. In addition regarding the future developments, the discretization of the α equation should be carried out so that it is blended in similar fashion as some of the modern schemes. Now it was compressed in every index direction, which is not necessary (in fact, the effects of this might be visible in Fig. 4.53, where there is a small but visible step in the contour $\alpha = 0.99$, as well as in the results obtained with the reference code). This could require more under-relaxation in the algorithm. However, it is important to remember that next to no difference was seen in the traditional definition $\alpha = 0.5$. The dynamic boundary condition then again was approximated slightly better using the more compressive discretization with FINFLO, and very accurately using YAFFA.

Many of the NVF type of schemes do not seek the higher-order formal accuracy, since this is not required when the distribution of α is not expected to be smooth in all coordinate directions (cf. Sec. 3.3.2 and Ubbink and Issa (1999)). Wackers (2007) discusses that in the numerical solutions

the spatial derivatives of α, which are not defined due to its discontinuous nature, do not converge upon grid refinement. The higher-order terms in the truncation error do not necessarily become smaller than the low-order terms. Then, seeking a higher-order scheme by eliminating the low-order truncation terms does not necessarily guarantee lower truncation error than the low-order schemes; thus, the fourth requirement in Sec. 2.2.2 could be partially omitted. Therefore, it is advisable[5] to base the discretization upon the angle γ as was discussed in Sec. 3.3.2, and further demonstrated in Sec. 4.1. Time-accurate simulations performed with YAFFA indicated that neglecting this might cause trouble. In steady-state situations, the interface always follows the velocity vector, and the angle γ can be obtained from this quite easily. Wackers (2007) mentions that the monotone discretization scheme is sufficient for reasonable prediction of the steady-state interface, since even the limited downwind scheme can only wrinkle the interface and not deform it. In unsteady cases, such as those computed in Sec. 4.1, the time stepping must be taken into account. This was found to be problematic in certain occasions, for if the time step is large enough, then the incoming fluxes can cause $\alpha > 1$ in a cell. In addition, the steady-state computations using FINFLO were done with constraining the α to its physical bounds. Ubbink and Issa (1999) note that they observed spurious velocity fields, originating from the conservation error that affected the momentum equations, when they reset the non-physical volume fraction values; they however focused on transient problems.

In search of a steady state, the time step acts as an under-relaxation parameter. The inherent dependency, because of either stability or accuracy, of the C in some of the schemes that were presented here is then a major disadvantage, usually obstructing the use of an optimum value for it. Several ways have been presented in the literature to deal with this. For example, Wackers et al. (2011) build half of their BICS to define the NVD such that it approaches the limited central difference scheme as the C is increased to prevent the instabilities of the more compressive schemes. Wackers (2007) designs a specific steady-state compressive scheme that blends between the upper limit of the TVD criterion and a third-order MUSCL scheme, based on familiar angle γ; in a steady-state case this angle is directly that of the

[5]Despite these, the SUPERBEE-limited scheme was found to perform best in the few problems that were computed here. This conclusion was based on different aspects, and those were already discussed. It should be kept in mind that only one type of unsteady problem was computed.

velocity vector. If a time integration is used to achieve the steady state, the recent bounded integration schemes of Moukalled and Darwish (2012) (for transient problems) show very promising results. Leroyer et al. (2009) present a temporal sub-cycling for the α equation by splitting the time step used for the bulk flow variables. Then, the α equation is solved sequentially several times within one global time step. Such method has been applied also at least by Wöckner-Kluwe (2013) and Berberović (2010). Berberović (2010) notes that a problem using this approach stems from the treatment of the fluxes during each sub-cycle. The implicit solution of FINFLO uses local time stepping, and already using the moving grid technique a different value for the pseudo-time step of the solution for the wave height than for the bulk flow variables is advisable (Meinander et al., 2006; Mikkola, 1999) when computing steady wave fields. Similar can be advised for the α equation, since now the code uses only one Courant number; this could have eluded some of the problems encountered in Sec. 4.2.

The treatment of gravity

In the solution of the flow equations with FINFLO, the pressure difference is used. But using the previous form (5.5), the working variable is explicitly p_{dif} and the casting of wall pressure boundary condition, as well as the still gravity source term in a quiescent fluid-fluid interface, on a single pressure system would be simplified (Rusche, 2002); another way to achieve the latter was addressed in Sec. 2.3 as well. YAFFA used a third means which was based on the work of Queutey and Visonneau (2007). The pressure profiles predicted by FINFLO are surprisingly good considering the fact that their calculation is based on the simple averages. The pressure difference is however not exactly the same by its definition as it is in YAFFA, for instance. This stems from the rather different treatment of the source term, given in Sec. 3.2.4, which is constant. That is, the hydrostatic part of the pressure remains constant despite the evolution of the interface. The total pressure is unaltered by this manipulation. This approach seemed to provide good results in the computed case. Note again that the interface that was predicted was not sharp.

Future work

There are auxiliary aspects that need to be dealt with, in addition to the remarks made above, if the model is to be applied to certain practical flow cases. Because of the types of problems that were solved here, many of these have been avoided. But for the sake of completeness, without being exhaustive, some important issues will be briefly addressed below.

Looking back to Sec. 1.2, in predictions of the resistance of a ship it is very important that the location of the free surface along the hull is estimated accurately. A great error can be made in the viscous resistance if the wetted surface is not approximated adequately, and additionally dissipation of kinetic energy is proportional to the square of the wave height. The modern moving grid techniques accomplish this with ease, but obvious difficulties were already discussed. However until grid convergence, it is unclear where the free surface is in simulations with a capturing strategy that uses the volume fractions. This can be remedied based on the study of the volume fraction equation that was presented in this thesis; also the mesh refinement criterion is rather simple. Another issue can rise due to the no-slip boundary condition. Especially when using small Reynolds number models, the $y^+ \approx 1$ on the hull of the ship which results in very thin computational cells with high aspect ratios. Then the local convection velocity is very small, and consequently the time evolution of the volume fraction is mild as well. This can result in a halted convergence close to the hull surface, and oscillations in the normal direction of it. A third problem, partially relating to the above, has been reported in applications of these capturing strategies to ship flow cases. Orych (2013) and Asén (2014) discuss that the risk of entrapment of the gas phase beneath the hull in these simulations is substantial. Such an occasion is depicted schematically in Fig. 5.2. It is clear that this behaviour is erroneous with regard to the physics of the situation, but the mono-fluid approach is still based on their simplifications, especially on the interface (cf. Sec. 1.1); moreover, now a third phase is present. In addition, no turbulence model is valid in this region, so the assumption of the complete phase segregation must hold. Orych (2013) solves this problem by extrapolating the water volume fraction a distance from the wall. With regard to the mixture model, the closer inspection of the two latter is left for a future study. It could also be based on an artificial diffusion term that is written to the α

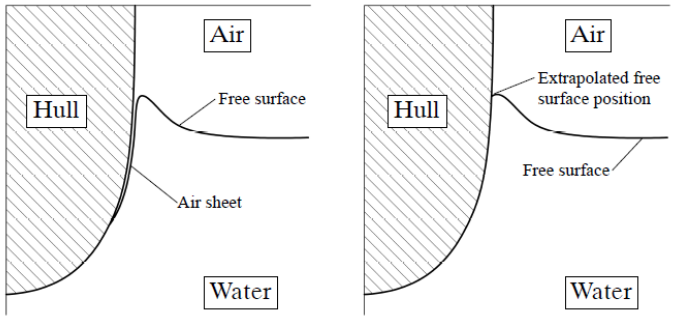


Figure 5.2: An example of the gas phase entrapment near the hull surface. Figure taken from Orych (2013).

equation (cf. Meinander et al. (2006)), or on a wall-oriented force.

The extension of the code to the other half of those given in Sec. 1.2 obliges a description for the water wave. A pressure-based isothermal solver has one additional variable, namely the volume fraction. The α equation is hyperbolic, so in an incompressible case also its value is needed at the inflow boundary. If we consider a linear wave progressing in the direction of the x-axis, the free surface elevation is of the form

$$\eta = A\ \mathsf{cos}(kx - \omega t)\ , \tag{5.8}$$

where A is the amplitude of the elevation, $k = 2\pi/\lambda$ the wave number and ω the angular frequency (Newman, 1977). Newman (1977) further shows that this elevation can be related to the local velocity components via the potential theory. Those can then be used for the inflow boundary to simulate the desired wave. The values for the volume fractions at the inflow boundary follow from the wave elevation, Eq. (5.8), as shown in Fig. 5.3. The pressure is a numerical condition at the inflow boundary, and can be extrapolated based on the distribution of α. Care must be taken in the simulation of the progressing wave over long distances, since the diffusion and dispersive type errors are naturally present. This type of an approach is not restricted to linear waves, but can be used with other wave theories such as irregular waves (Düz et al., 2011; Paulsen, 2013), Stokes waves (Vaz et al., 2009), cnoidal and solitary waves, as mentioned by Lin (2008).

The boundary conditions at the outflow boundaries are not as straight-

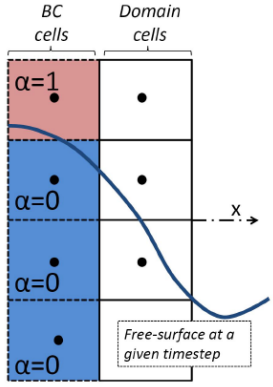


Figure 5.3: A sketch of a wave inflow boundary condition. Figure taken from Vaz et al. (2009).

forward. One immediately can think of the similar wave theory to account for the pressure via the Bernoulli relation. This would be valid for an undisturbed wave; Wöckner et al. (2007) use a similar approach but with the velocity components, and show that this results in a phase error of the progressive wave. The present case produced waves in the initially undisturbed domain, and these progressed toward the outlet[6]; that is, there is no way of knowing the solution past a finite downstream boundary of the domain. Due to the nature of the equations, usually the pressure boundary condition is used as the sole physical one at the downstream boundary. The typical way to deal with these at the outlet is the numerical beach approach, in which the numerical diffusion is used to fade any disturbance before they reach the domain boundary, and the initial state prevails. In this study, too fine grids were used, and the problems stemming from this had been discussed. Below, a few possible solutions are considered.

A widely referenced alternative is the Sommerfeld radiation boundary condition, cf. Lin (2008), Düz et al. (2011), Wellens (2012) and Wöckner-Kluwe (2013). This is

$$\left(\frac{\partial}{\partial t} + c\frac{\partial}{\partial x}\right) U = 0 \ , \tag{5.9}$$

[6]It is reminded that upstream advancing solitons invoked reflections at the inflow boundary as well; this can take place also if there is a solid body inside the domain. Lin (2008) presents a discussion for dealing with these, similar to the Sommerfeld condition.

where c is the local wave phase speed and U is any flow variable (Lin, 2008). Condition of this type allow the majority of the waves to leave the domain without significant reflection. Düz et al. (2011) formulate a similar relation for absorbing irregular waves. Wellens (2012) present a good discussion of these types of boundary conditions. A similar condition can be arrived at based on the nature of the flow equations. If we consider a hyperbolic system, it is possible to present the equations in a characteristic form. Those equations consist of waves of the characteristic variables that are propagated with the characteristic velocities, which are the eigenvalues of the hyperbolic system; for the equations used in YAFFA, these are the flow velocity components. Using the characteristic form, waves in each characteristic variable can be considered separately at the domain boundaries, as described by Thompson (1987, 1990). This way, we can require that the waves in these characteristic variables vanish at the domain boundaries, and efficient non-reflecting conditions can be constructed using the primitive variables. Wellens (2012) applies the method of characteristic to the wave equation $\phi_{tt} - c^2\phi_{xx}$, with ϕ being the velocity potential, which can be factored to the same form. For progressive waves, the phase velocity must follow from local flow convection velocity, as stated by the Rankine-Hugoniot conditions for the propagation of the contact discontinuity; note that this presents a non-linearity.

A third approach is detailed by Wöckner-Kluwe (2013). It is based on an implicit coupling of an inviscid wave description in the far field to the viscous flow in the interior of the domain. She manipulates the linear system by modifying the diagonal coefficient and the source term, interpreted to the present notation, as

$$\left(1 + \beta\alpha^m\right) \mathsf{A}_P U_P^{n+1} + \sum_{nb} \mathsf{A}_{nb} U_{nb}^{n+1} = R + \mathsf{A}_P \beta\alpha^m U_P^* \,, \tag{5.10}$$

where the new term β controls the intensity of this manipulation and α^m is now defined as a shape function (to the power of m); the matrix coefficients are given by Eq. (3.30) for the momentum equations and in Eq. (3.94) for the gas mass fraction; the summation is taken over the cells neighbouring (nb) the cell P; U denotes either the velocity or the gas mass fraction. In application of this method, the changes of these variables were not used by Wöckner-Kluwe (2013); this has been the present practice. However, if we assume the solution on the current iteration level already satisfies these linear equations, they can be written solely in terms of these variables on

the new level. The shape function restricts this manipulation to prescribed regions near the inflow and outflow boundaries; it gets the value of $\alpha = 1$ at these boundaries and $\alpha = 0$ in the domain outside the zone of $\lambda/2$ from the boundaries, being linear ($m = 1$) or quadratic ($m = 2$) in this zone. This manipulation results in $U_P \approx U_P^*$ for large values of $\beta\alpha^m$. The $U_P^* = (\alpha, u, v, w)$ are known, determined from the inviscid method such as the above linear wave theory. β is the most crucial parameter in this strategy. Using this implicit approach and a ship in head waves as an example, the outflow boundary can be placed at only $2/3 \cdot L_{pp}$, where the L_{pp} is the length between perpendiculars, behind the ship (Wöckner-Kluwe, 2013); usually, a successful computation entails this distance to be an order of magnitude larger.

The α equation can also be written in an alternative form. This has not been addressed earlier in this thesis, but it is nonetheless worthwhile to present here since it provides interesting features. The form originates from the 'counter-gradient' transport that plays an important role in the dynamics of turbulent flames in complex combustion models (Weller, 2008). It is based on a single expression of the convection equation for α that still possesses conservative and bounded behaviour

$$\frac{\partial \alpha}{\partial t} + \frac{\partial \alpha u_i}{\partial x_i} + \frac{\partial \alpha(1-\alpha)u_{i,c}}{\partial x_i} = 0 \ , \tag{5.11}$$

where $u_{i,c}$ is the 'compression velocity'. It corresponds to the relative velocity of the burnt and un-burnt gases (Weller, 2008). The artificial compression term is only active in the region of the interface because of the non-linear term. The benefit of this is the evasion of the specific discretization strategy for the traditional convection term, the only requirement being a bounded version since the compression comes from this additional term. Bounded schemes were readily introduced in this work, and such can be constructed from practically any of the TVD or NVF discretization strategies for the cell faces value of α in the control volume equivalent of Eq. (5.11). However, the nature of the compression term is not a convective one but rather 'counter-diffusive'. For this reason, even an upwind scheme produces decent results, and Weller (2008) suggests a weighed blending between central differencing and the upwind scheme using a suitable 'limiter' for this weighing. In a homogeneous system the compression velocity must be modelled, and a general form suggested by Weller (2008) is $u_{i,c} = \mathsf{min}\Big(c_\alpha |u_i|, \mathsf{max}(|u_i|)\Big) \nabla\alpha / |\nabla\alpha|$,

with c_α being a compression coefficient for the scheme, which should be of the order 1, and $\mathsf{max}(|u_i|)$ now the maximum value for the speed in the domain. This discretization strategy has been applied at least by Rusche (2002)[7], Berberović (2010), Kissling et al. (2010), Paulsen (2013) and Asén (2014). Berberović (2010) performed studies varying the coefficient c_α, which showed that decent results can be achieved even with $c_\alpha = 0$, and that the interface becomes more confined, ultimately between two cells, as it is increased $c_\alpha \to 1$. All workers show that good results can be achieved from using Eq. (5.11) in practical flow cases. Moreover, Paulsen (2013) and Asén (2014) applied this for numerical simulations of wave impacts on offshore structures and for flows around a cruise ferry, respectively, by using the OpenFOAM solver.

Conclusion

It is evident that an efficient free surface capturing strategy that uses the VOF method must reflect to almost all parts of the solution algorithm. Additionally, there is not a single method of solution that can be raised above every other. By following certain guidelines, an effective algorithm can be constructed. Few of those guidelines have been addressed in this study. The existence of some others have been acknowledged in this chapter.

One goal of this study was to gather the state-of-the-art of the aforesaid parts. The present VOF model, the mixture model, is not in its entirety completed, and further developments are inevitably involved with those aspects that were studied and discussed within this thesis. The issues that were found to be important within this work have been discussed earlier in this chapter. For example, if one considers any of the flow cases introduced in Chap. 1, the discretization of the void fraction equation must be based on another and more suitable method; it faces, in principle, the same requirements that were discussed in Chaps. 2, 3 and 4 in all cases.

In the above context, a built-in aspiration was also to track and detect the deficiencies or differences of the current algorithm compared to what would impose better applicability to the free surface problem, in addition to

[7]He refers to private communications with Henry Weller in the description of this discretization.

those used by others. The solution methods were implemented to separate codes which makes it impossible to investigate all of them in one master's thesis, but it proved fruitful to consider some of the most important ones on a more abstract, or general, level. Then, it is possible to establish the verification, and indeed, such a conclusion can be made. If one returns to the definition of the process of verification, it is possible to conclude that the discrete mathematics model, the mixture model, applies the conceptual model, which was here the problem of a free surface present between two immiscible and segregated fluids. But as was noted above, it is not complete. Henceforth, the previously presented guidelines should be followed.

Bibliography

P. Asén. Analysis of the flow around a cruise ferry hull by the means of computational fluid dynamics. Master's thesis, Aalto University, School of Engineering, 2014.

E. Berberović. *Investigation of free-surface flow associated with drop impact: numerical simulations and theoretical modeling.* PhD thesis, Technischen Universität Darmstadt, November 2010.

F. Bet, D. Hänel, and S.D. Sharma. Simulation of hydrodynamical free-surfece flows. *ECCOMAS'96, John Wiley & Sons Ltd.*, pages 259 – 277, 1996.

S. Bhushan, T. Xing, P. Carrica, and F. Stern. Model- and full-scale URANS simulations of Athena resistance, powering, seakeeping, and 5415 maneuvering. *J. Ship Res.*, 53(4):179 – 198, 2009.

P.M. Carrica, R.V. Wilson, and F. Stern. An unsteady single-phase level set method for viscous free surface flows. *IIHR Technical Report No. 444*, 2005.

M. Darwish. A new high-resolution scheme based on the normalized variable formulation. *Num. Heat Trans. B*, 24(3):353 – 371, 1993.

M. Darwish and F. Moukalled. Convective schemes for capturing interfaces of free-surface flows on unstructured grids. *Num. Heat Trans. B*, (49): 19–42, 2006.

S.F. Davis. Flux difference splittings and limiters for the resolution of contact discontinuities. *Appl. Math. Comp.*, (65):3–18, 1994.

B. Düz, R.H.M. Huijsmans, A.E.P. Veldman, M.J.A. Borsboom, and P.R. Wellens. An absorbing boundary condition for regular and irregular wave simulations. In L. Eça, E. Oñate, J. García-Espinosa, T. Kvamsdal, and P. Bergan, editors, *MARINE2011, IV International conference on computational methods in marine engineering. Selected papers*, 2011.

L. Eça, E. Oñate, J. García-Espinosa, T. Kvamsdal, and P. Bergan. MARINE2011, IV international conference on computational methods in marine engineering. Selected papers. 2011.

J.H. Ferziger and M. Perić. *Computational methods for fluid dynamics.* Springer, New York, second edition, 1999.

Finflo. FINFLO *user guide.* Finflo Ltd., 2013.

C.A.J. Fletcher. *Computational techniques for fluid dynamics, Vol. 1.* Springer, New York, second edition, 1991.

Wei Gao, Hong Li, Yang Liu, and Yong-Jun Jian. An oscillation-free high order TVD/CBC–based upwind scheme for convection discretisation. *Numer. Algor.*, 59:29–50, 2012.

P.H. Gaskell and A.K.C. Lau. Curvature-compensated convective transport: SMART, a new boundedness-preserving transport algorithm. *Int. J. Num. Meth. Fluids*, 8:617 – 641, 1988.

A. Harten. High resolution schemes for hyperbolic conservation laws. *J. Comp. Phys.*, (49):357 – 393, 1983.

D.P. Hill. *The computer simulation of dispersed two-phase flows.* PhD thesis, Imperial College of Science, Technology & Medicine, 1998.

T. Hino. CFD workshop. Tokyo. 2005.

C.W. Hirt and B.D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comp. Phys.*, (39):201–25, 1981.

S. Hänninen and T. Mikkola. On the implementation of an interface capturing method. 10^{th} *numerical towing tank symposium*, September 2007.

S. K. Hänninen. *Second harmonic wave loads as springing excitation of a large cruise ship in short and steep head waves.* PhD thesis, Aalto University, School of Engineering, 2014.

M. Hoekstra, G. Vaz, B. Abeil, and T. Bunnik. Free-surface flow modelling with interface capturing techniques. In P. Bergan, J. Garcia, E. Oñate, and T. Kvamsdal, editors, *MARINE2007, International conference on computational methods in marine engineering.*, 2007.

W.-L. Hong and D.T. Walker. Reynolds-averaged equations for free-surface flows with application to high-Froude-number jet spreading. *J. Fluid Mech.*, 417:183 – 209, 2000.

M.E. Hubbard and P. Carcia-Navarro. Flux difference splitting and the balancing of source terms and flux gradients. *J. Comp. Phys.*, 165, 2000.

H. Jasak. *Error analysis and estimation for the finite volume method for application to fluid flows.* PhD thesis, Imperial College of Science, Technology & Medicine, 1996.

H. Jasak and H. G. Weller. Interface tracking capabilities of the Inter-Gamma Differencing Scheme. Imperial College of Science, Technology & Medicine, 1995.

H. Jasak, H.G. Weller, and A.D. Gosman. High resolution NVD scheme for arbitrarily unstructured meshes. *Int. J. Num. Meth. Fluids*, (31):431 – 449, 1999.

K. Kissling, J. Springer, H. Jasak, S. Schütz, K. Urban, and M. Piesche. A coupled pressure based solution algorithm based on the volume-of-fluid approach for two or more immiscible fluids. In J.C.F. Pereira and A. Sequeira, editors, *V European Conference on Computational Fluid Dynamics ECCOMAS CFD*, 2010.

T. Kukkanen. *Numerical and experimental studies of nonlinear wave loads of ships.* PhD thesis, VTT Technical Research Center of Finland, 2012.

B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti. Modelling merging and fragmentation in multiphase flows with SURFER. *J. Comp. Phys.*, 113, 1994.

L. Larsson and H.C. Raven. *Ship resistance and flow.* The society of naval architects and marine engineers, New Jersey, first edition, 2010.

L. Larsson, F. Stern, and V. Bertram. A workshop on numerical ship hydrodynamics. Gothenburg. 2000.

L. Larsson, F. Stern, and M. Visonneau. CFD in ship hydrodynamics. Results from the Gothenburg 2010 workshop. 2010.

B.P. Leonard. Simple high-accuracy resolution program for convective modelling of discontinuities. *Int. J. Num. Meth. Fluids*, 8:1291 – 1318, 1988.

B.P. Leonard. The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection. *Comp. Methods Appl. Mech. Engng.*, (88):17 – 74, 1991.

A. Leroyer, P. Queutey, E. Guilmineau, G. Deng, and M. Visonneau. New algorithms to speed up RANSE computations in hydrodynamics. In V. Bertram, editor, *Numerical Towing Tank Symposium NuTTS'09*, 2009.

R. J. LeVeque. Numerical methods for conservation laws. Lectures in mathematics, ETH Zürich, 1992.

R.J. Leveque. Balancing source terms and flux gradients in high-resolution Godunov methods: the quasi-steady wave-propagation algorithm. *J. Comp. Phys.*, 146, 1998.

T.-Q. Li and J. Matusiak. Simulation of viscous flow of modern surface ships using the FINFLO RANS solver. In *Practical Design of Ships and Other Floating Structures. Proceedings of the Eighth International Symposium on Practical Design of Ships and Other Floating Structures*, Shanghai, China, 16-21 September 2001.

T.-Q. Li, J. Matusiak, and R. Lehtimäki. Numerical simulation of viscous flows with free surface around realistic hull forms with transom. *Int. J. Numer. Meth. Fluids*, 37, 2001.

P. Lin. *Numerical modeling of water waves.* Taylor & Francis, London, first edition, 2008.

K. Lowery and S. Liapis. Free-surface flow over a semi-circular obstruction. *Int. J. Numer. Meth. Fluids*, 30:43–63, 1999.

J. Matusiak. *Johdatus laivan aallonmuodostukseen.* M-227. Otaniemi, Espoo, Finland, third edition, 2005. (in Finnish).

M. Meinander, E. Salminen, and T. Siikonen. Implementation of a free surface model into FINFLO code. Technical report F-37. Finflo Oy, 2006.

A. Miettinen, E. Salminen, and T. Siikonen. Kaksifaasivirtaussimulointimenetelmien kehitystä ja sovelluksia. Technical report F-10. Finflo Oy, 2006. (in Finnish).

T. Mikkola. Testing of two FINFLO -based free-surface codes with Eulerian flow over a Gaussian ground elevation. Report D-56. Helsinki University of Technology, 1999.

T. Mikkola. *Simulation of unsteady free surface flows - code verification and discretisation error.* PhD thesis, Helsinki University of Technology, 2009.

H. Montazeri. *A consistent numerical method for simulating interfacial turbulent flows.* PhD thesis, University of Toronto, 2010.

H. Montazeri, M. Bussmann, and J. Mostaghimi. Accurate implementation of forcing terms in for two-phase flows into SIMPLE algorithm. *Int. J. Multiphase Flow*, (45), 2012.

H. Montazeri, M. Bussmann, and J. Mostaghimi. A balanced-force algorithm for two-phase flows. *J. Comp. Phys.*, (257), 2014.

F. Moukalled and M. Darwish. Transient schemes for capturing interfaces of free-surface flows. *Num. Heat Trans. B*, 61(3):171 – 203, 2012.

S. Muzaferija and M. Perić. Computation of free-surface flows using the finite-volume method and moving grids. *Num. Heat Transf. B*, 4(32):369 – 384, 1997.

K. Nandi and A.W. Date. Formulation of fully implicit method for simulation of flows with interfaces using primitive variables. *Int. J. Heat Mass Transf.*, 52(32):3217 – 3224, 2009.

J.N. Newman. *Marine hydrodynamics.* The MIT Press, Cambridge, eighth edition, 1977.

W.L. Oberkampf and T.G. Trucano. Verification and validation in computational fluid dynamics. *Prog. Aerosp. Sci.*, 38, 2002.

M. Orych. *Development of a free surface capability in a RANS solver with coupled equations and overset grids.* Licentiate thesis, Chalmers University of Technology, 2013.

S. V. Patankar. *Numerical heat transfer and fluid flow.* Hemisphere publishing corp., New York, 1980.

B.T. Paulsen. *Efficient computations of wave loads on offshore structures.* PhD thesis, The Technical University of Denmark, 2013.

W.H. Press, W.T. Vetterlig, S.A. Teukolsky, and B.P. Flannery. *Numerical recipes in Fortran.* Cambridge University Press, New York, second edition, 1992.

L. Qian, D.M. Causon, C.G. Mingham, and D.M. Ingram. A free-surface capturing method for two fluid flows with moving bodies. *Proc. R. Soc. A*, 462, 2006.

P. Queutey and M. Visonneau. An interface capturing method for free-surface hydrodynamic flows. *Comp. Fluids.*, 36:1481–1510, 2007.

S.H. Rhee, B.P. Makarov, H. Krishinan, and V. Ivanov. Assessment of the volume of fluid method for free surface wave flow. *J. Marin. Sci. Technol.*, (10), 2005.

M. Rudman. Volume-tracking methods for interfacial flow calculations. *Int. J. Num. Meth. Fluids*, 24, 1997.

H. Rusche. *Computational fluid dynamics of dispersed two-phase flows at high phase fractions.* PhD thesis, Imperial College of Science, Technology & Medicine, 2002.

H. Sadat-Hosseini, P. Carrica, F. Stern, N. Umeda, H. Hashimoto, S. Yamamura, and A. Mastuda. CFD, system-based and EFD study of ship dynamic instability events; surf-riding, periodic motion, and broaching. *Ocean Engineering*, 38:88 – 110, 2011.

D. Schmonde, K. Wöckner, and T. Rung. Free surface modelling in FRESCO, latest developments. In V. Bertram, editor, *Numerical Towing Tank Symposium NuTTS'07*, 2007.

J. Schweighofer. *Investigation of two-dimensional transom waves using inviscid and viscous free-surface boundary conditions at model and full-scale ship Reynolds number.* PhD thesis, Helsinki University of Technology, June 2003.

A. Shiri, M. Leer-Andersen, R.E. Bensow, and J. Norrby. Numerical simulation of free surface break-up and re-attachment inside an air cavity. In R.E. Bensow, editor, *Numerical Towing Tank Symposium NuTTS'14*, 2014.

T. Siikonen. Numerical method for one-dimensional two-phase flow. *Num. Heat Trans.*, 12:1–18, 1987.

T. Siikonen. Laskennallisen virtausmekaniikan ja lämmönsiirron perusteet. Lecture notes in Aalto University course Ene-39.4037 Fundamentals in computational fluid dynamics and heat transfer, 2008. (in Finnish).

T. Siikonen. Developments of the cavitation model of the FINFLO code. Technical report F-44. Finflo Oy, 2009.

T. Siikonen. Developments in pressure correction methods for single and two phase flow. Aalto University, CFD-Group, September 2011. Memo: CFD/MECHA-10-2011.

T. Siikonen. Laskennallisen virtausmekaniikan ja lämmönsiirron jatkokurssi. Lecture notes in Aalto University course Ene-39.4030 Advances in computational fluid dynamics and heat transfer, 2013. (in Finnish).

T. Siikonen, J. Ilkko, and E. Salminen. Developments of the two-phase model for the FINFLO code. Technical report F-79. Finflo Oy, 2012.

T. Sipilä. *RANS analyses of cavitating propeller flows.* Licentiate thesis, Aalto University, September 2012.

F. Stern, J. Yang, Z. Wang, H. Sadat-Hosseini, M. Mousaviraad, S. Bhushan, and T. Xing. Computational ship hydrodynamics: nowadays and way forward. 29$^{\text{th}}$ *Symposium on Naval Hydrodynamics*, August 26-31 2012.

H.B. Stewart and B. Wendroff. Two-phase flow: models and methods. *J. Comp. Phys.*, 56:363–409, 1984.

P.K. Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J. Num. Anal.*, 21(5):995–1011, 1984.

J.C. Tannehill, D.A. Anderson, and R.H. Pletcher. *Computational fluid mechanics and heat transfer.* Taylor & Francis, Philadelphia, second edition, 1997.

K.W. Thompson. Time dependent boundary conditions for hyperbolic systems. *J. Comp. Phys.*, (68):1–24, 1987.

K.W. Thompson. Time dependent boundary conditions for hyperbolic systems, II. *J. Comp. Phys.*, (89):439–461, 1990.

O. Ubbink. *Numerical prediction of two fluid systems with sharp interfaces.* PhD thesis, Imperial College of Science, Technology & Medicine, 1997.

O. Ubbink and R.I. Issa. A method for capturing sharp fluid interfaces on arbitrary meshes. *J. Comp. Phys*, 153, 1999.

M. van Dyke. *An album of fluid motion.* The Parabolic Press, Stanford, California, 1982.

G. Vaz, F. Jaouen, and M. Hoekstra. Free-surface viscous flow computations. Validation of URANS code FRESCO. In *Proceedings of ASME* 28$^{\text{th}}$ *International conference on ocean, offshore and arctic engineering (OMAE)*, 2009.

V. Viitanen. Application of a homogeneous mixture model to the free surface problem. In R.E. Bensow, editor, *Numerical Towing Tank Symposium NuTTS'14*, 2014.

M. Visonneau. Physical and numerical modeling of turbulent incompressible flows. Application to viscous ship hydrodynamics. Lectures in Ship hydrodynamics, Aalto University, 2013.

J. Wackers. *Surface capturing and multigrid for steady free-surface water flows.* PhD thesis, Delft University of Technology, 2007.

J. Wackers and B. Koren. A surface capturing method for the efficient computation of steady water waves. *J. Comp. Appl. Math.*, (215), 2005.

J. Wackers, B. Koren, H.C. Raven, A. van der Ploeg, A.R. Starke, G.B. Deng, P. Queutey, M. Visonneau, T. Hino, and K. Ohashi. Free-surface viscous flow solution methods for ship hydrodynamics. *Arch. Comput. Methods Eng.*, 18:1–41, 2011.

N.P. Waterson and H. Deconinck. Design principles for bounded higher-order convection schemes – a unified approach. *J. Comp. Phys.*, 224, 2007.

K. Wöckner, P. Soukup, and T. Rung. Boundary conditions for free surface flow. In V. Bertram, editor, *Numerical Towing Tank Symposium NuTTS'07*, 2007.

K. Wöckner-Kluwe. *Evaluation of the unsteady propeller performance behind ships in waves.* PhD thesis, Technischen Universität Hamburg-Harburg, April 2013.

Jin-Jia Wei, Bo Yu, Wen-Quan Tao, Yasuo Kawaguchi, and Huoang-Sheng Wang. A new high-order-accurate and bounded scheme for incompressible flow. *Numer. Heat Transf. B*, 43(1):19–41, 2003.

P. R. Wellens. *Wave simulation in truncated domains for offshore applications*. PhD thesis, Technische Universiteit Delft, January 2012.

H.G. Weller. Derivation, modelling and solution of the conditionally averaged two-phase flow equations. Technical report, Open∇CFD, 2005.

H.G. Weller. A new approach to VoF-based interface capturing methods for incompressible and compressible flow. Technical report, Open∇CFD, 2008.

F.M. White. *Viscous fluid flow*. McGraw-Hill, New York, third edition, 2006.

H.C. Yee. A class of high-resolution explicit and implicit shock-capturing methods. *NASA Technical Memorandum 101088*, 1989.

B. Yu, W.Q. Tao, D.S. Zhang, and Q.W. Wang. Discussion on numerical stability and boundedness of convective discretised scheme. *Numer. Heat. Transf. B*, 40(4):343–365, 2001.

Appendices

Appendix A

Results from the convection of the step profile in an oblique velocity field

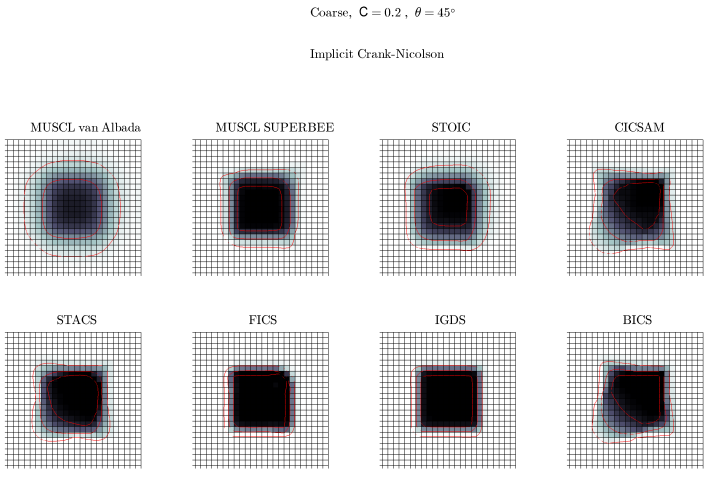


Figure A.1: Results from the square case with $\mathsf{C} = 0.2$ and $\theta = 45°$, coarse grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

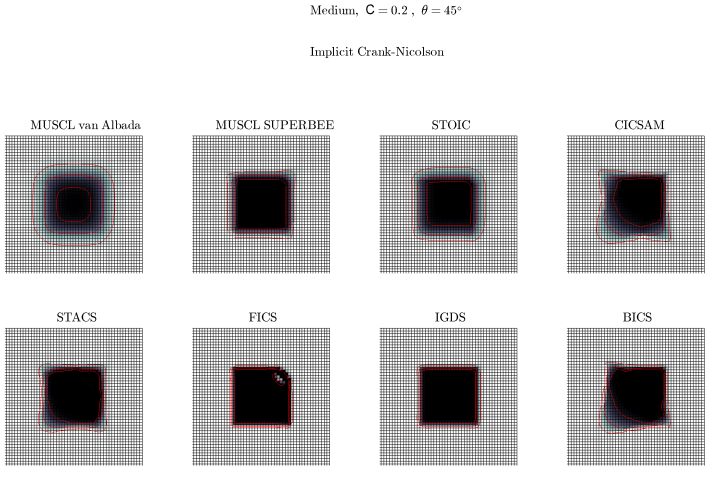


Figure A.2: Results from the square case with $\mathsf{C} = 0.2$ and $\theta = 45°$, medium grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

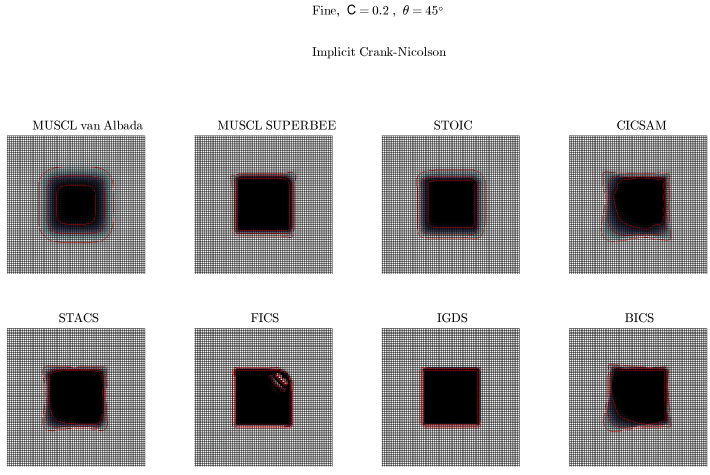


Figure A.3: Results from the square case with $\mathsf{C} = 0.2$ and $\theta = 45°$, fine grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

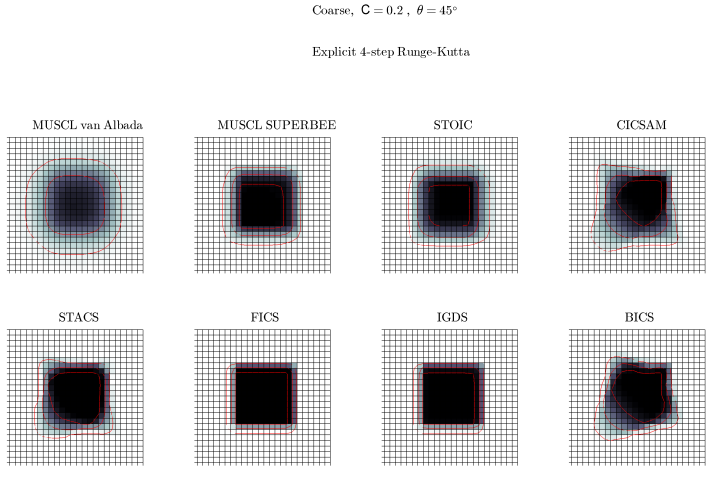


Figure A.4: Results from the square case with $\mathsf{C} = 0.2$ and $\theta = 45°$, coarse grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

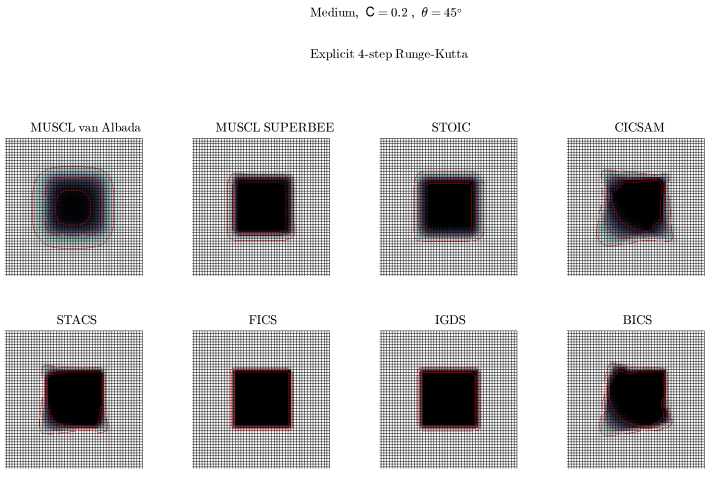


Figure A.5: Results from the square case with $\mathsf{C} = 0.2$ and $\theta = 45°$, medium grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

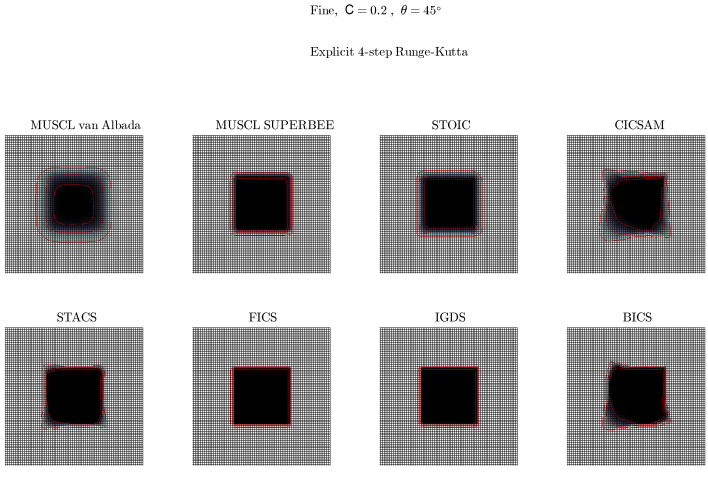


Figure A.6: Results from the square case with $\mathsf{C} = 0.2$ and $\theta = 45°$, fine grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

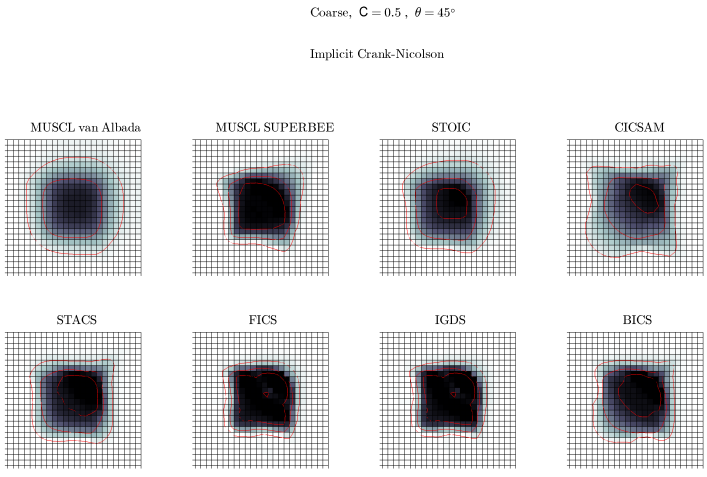


Figure A.7: Results from the square case with $\mathsf{C} = 0.5$ and $\theta = 45°$, coarse grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

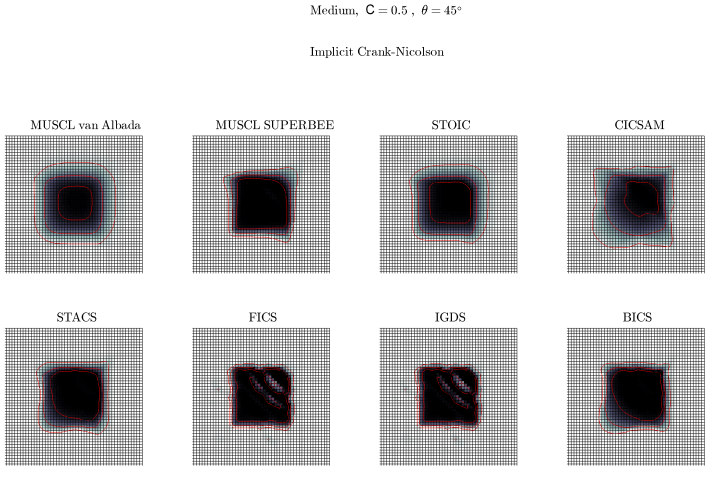


Figure A.8: Results from the square case with $\mathsf{C} = 0.5$ and $\theta = 45°$, medium grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

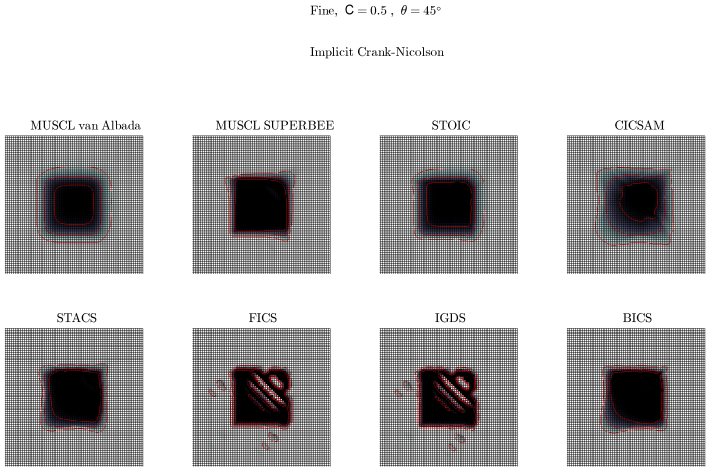


Figure A.9: Results from the square case with $\mathsf{C} = 0.5$ and $\theta = 45°$, fine grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

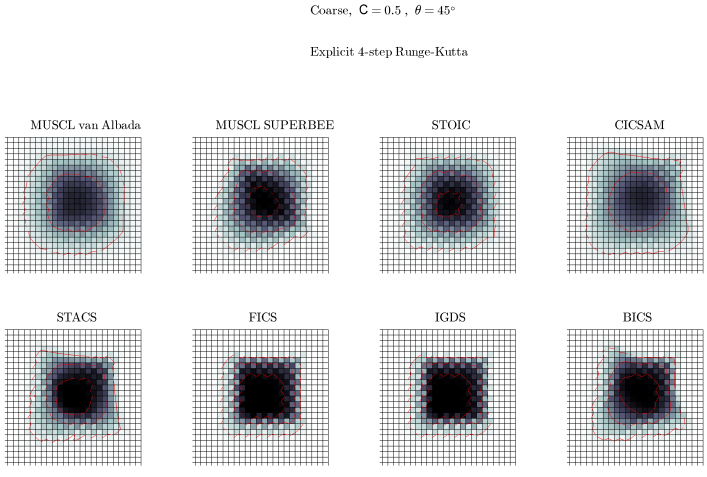


Figure A.10: Results from the square case with $\mathsf{C} = 0.5$ and $\theta = 45°$, coarse grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

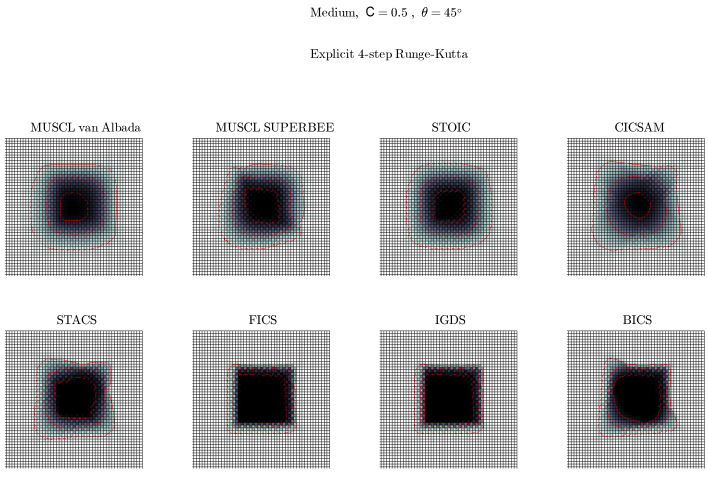


Figure A.11: Results from the square case with $\mathsf{C} = 0.5$ and $\theta = 45°$, medium grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

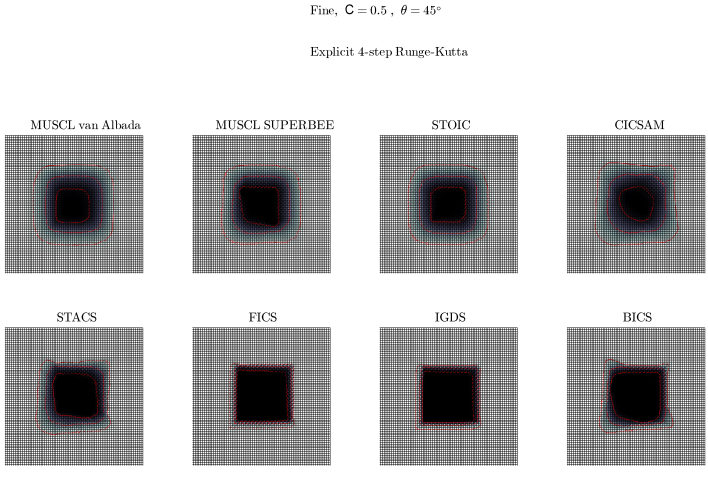


Figure A.12: Results from the square case with $\mathsf{C} = 0.5$ and $\theta = 45°$, fine grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

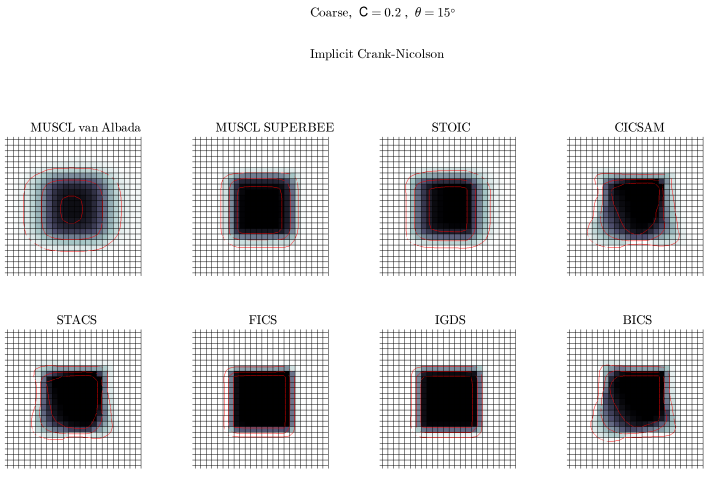


Figure A.13: Results from the square case with $\mathsf{C} = 0.2$ and $\theta = 15°$, coarse grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

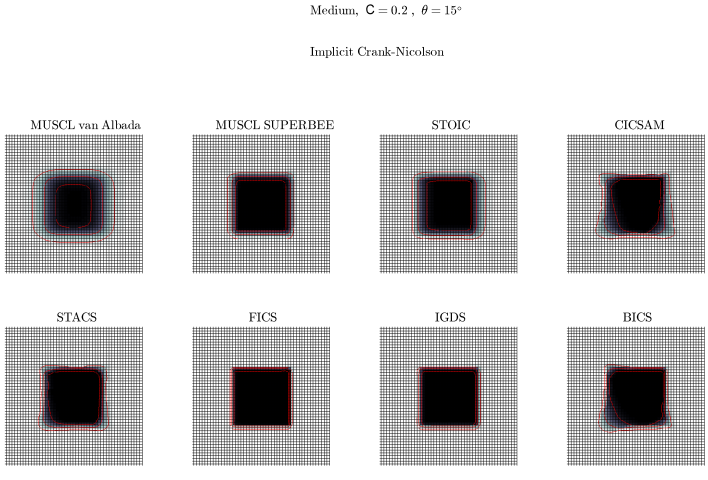


Figure A.14: Results from the square case with $\mathsf{C} = 0.2$ and $\theta = 15°$, medium grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

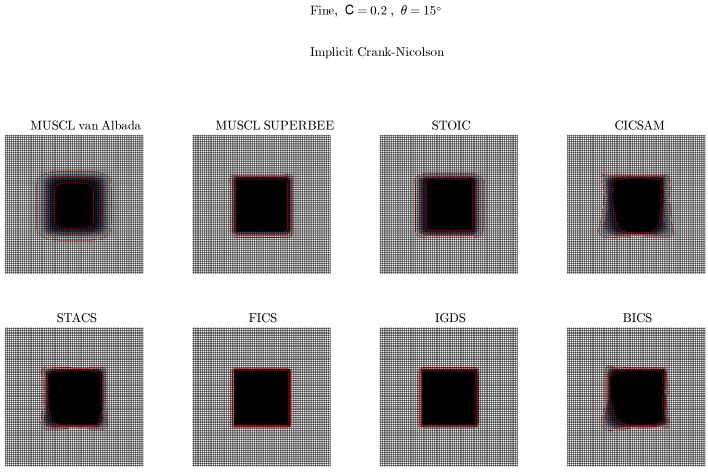


Figure A.15: Results from the square case with $\mathsf{C} = 0.2$ and $\theta = 15°$, fine grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

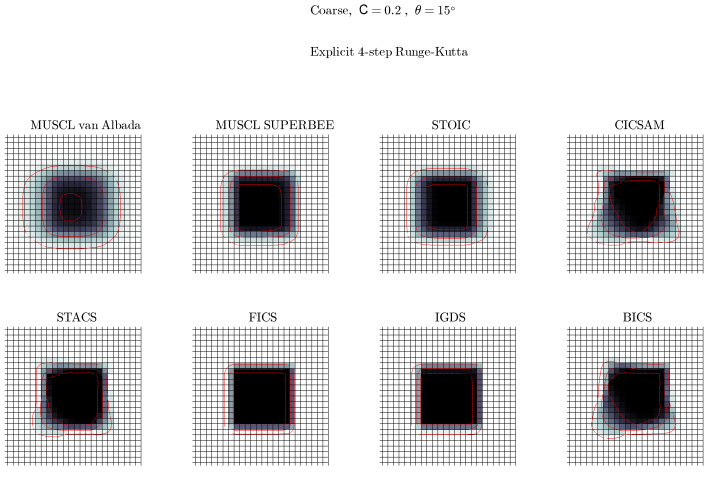


Figure A.16: Results from the square case with $\mathsf{C} = 0.2$ and $\theta = 15°$, coarse grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

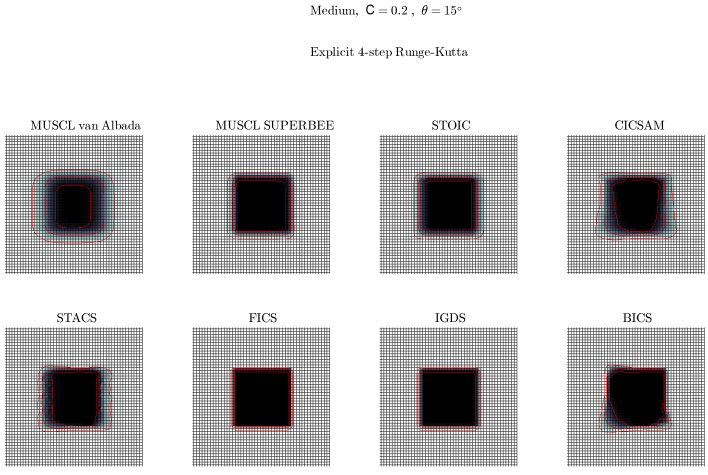


Figure A.17: Results from the square case with $\mathsf{C} = 0.2$ and $\theta = 15°$, medium grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

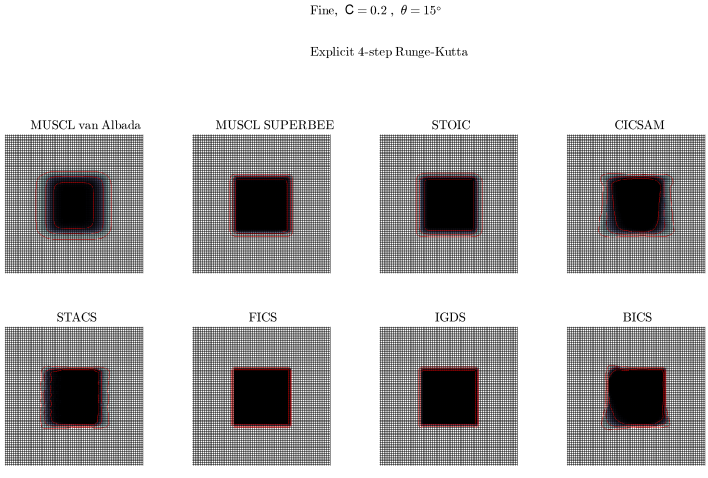


Figure A.18: Results from the square case with $\mathsf{C} = 0.2$ and $\theta = 15°$, fine grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

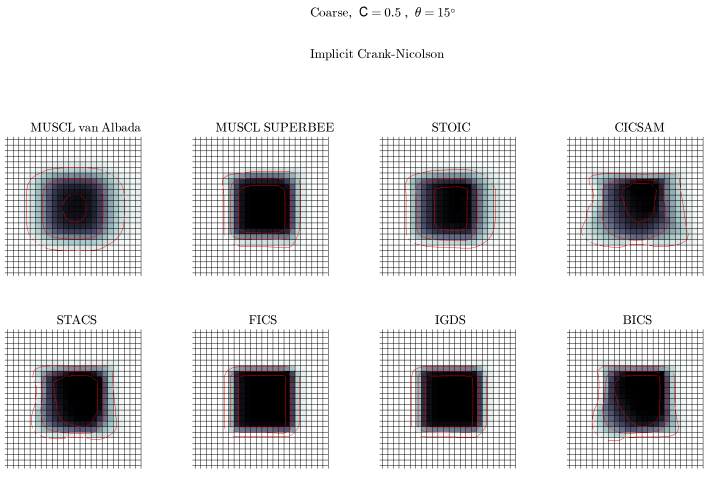


Figure A.19: Results from the square case with $\mathsf{C} = 0.5$ and $\theta = 15°$, coarse grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

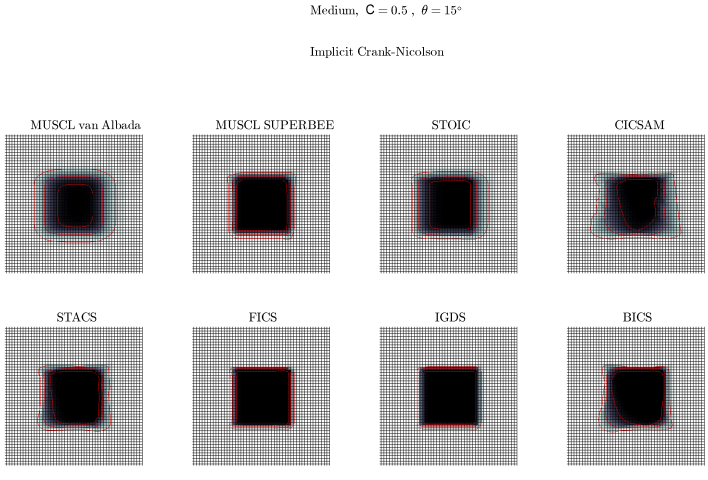


Figure A.20: Results from the square case with $\mathsf{C} = 0.5$ and $\theta = 15°$, medium grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

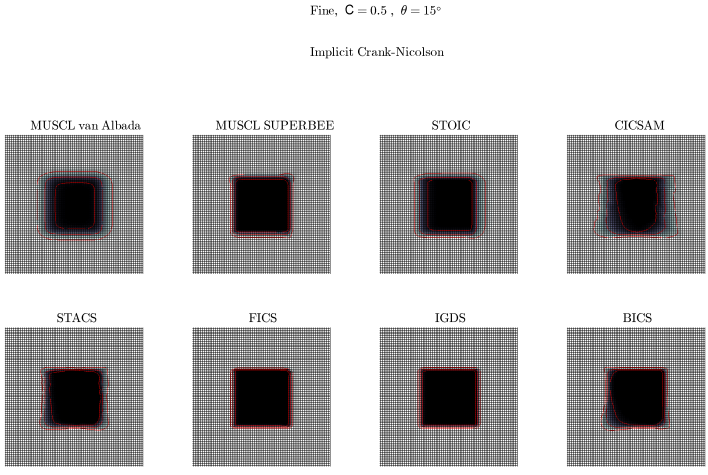


Figure A.21: Results from the square case with $\mathsf{C} = 0.5$ and $\theta = 15°$, fine grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

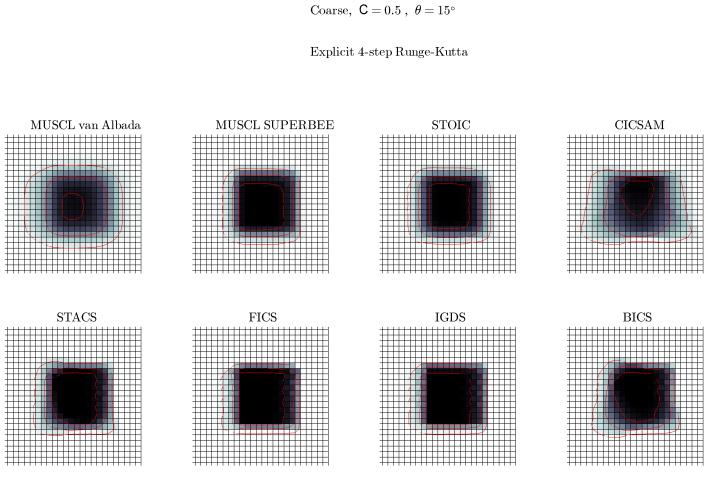


Figure A.22: Results from the square case with $\mathsf{C} = 0.5$ and $\theta = 15°$, coarse grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

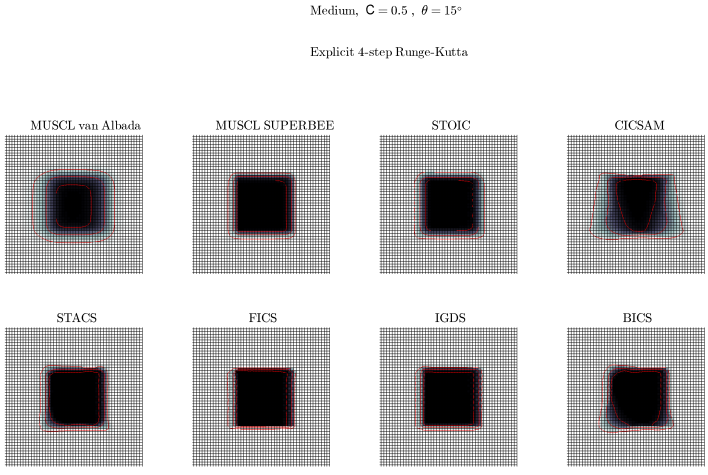


Figure A.23: Results from the square case with $\mathsf{C} = 0.5$ and $\theta = 15°$, medium grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

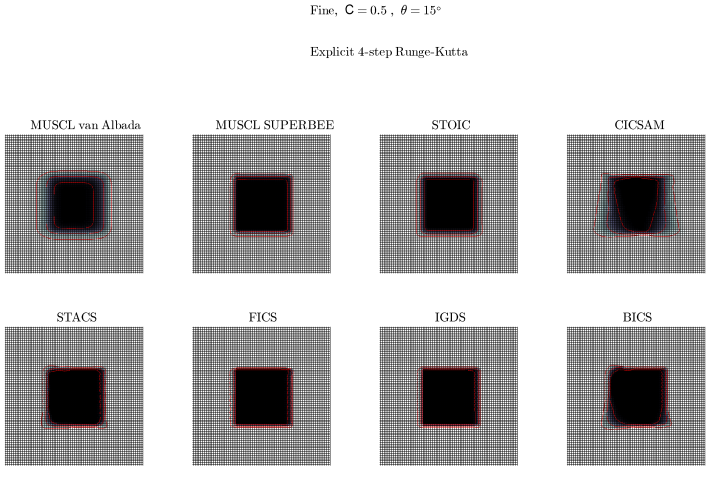


Figure A.24: Results from the square case with $\mathsf{C} = 0.5$ and $\theta = 15°$, fine grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

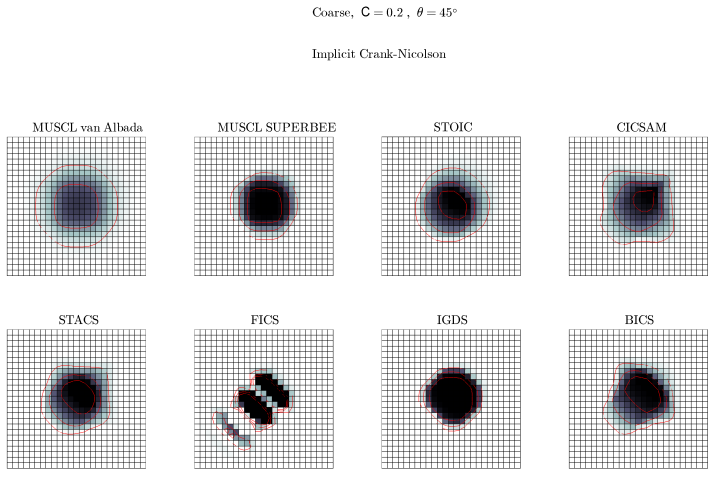


Figure A.25: Results from the circle case with $\mathsf{C} = 0.2$ and $\theta = 45°$, coarse grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

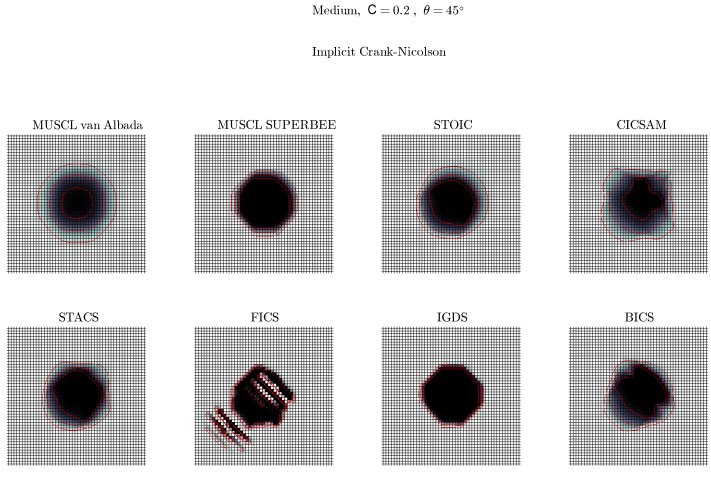


Figure A.26: Results from the circle case with $\mathsf{C} = 0.2$ and $\theta = 45°$, medium grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

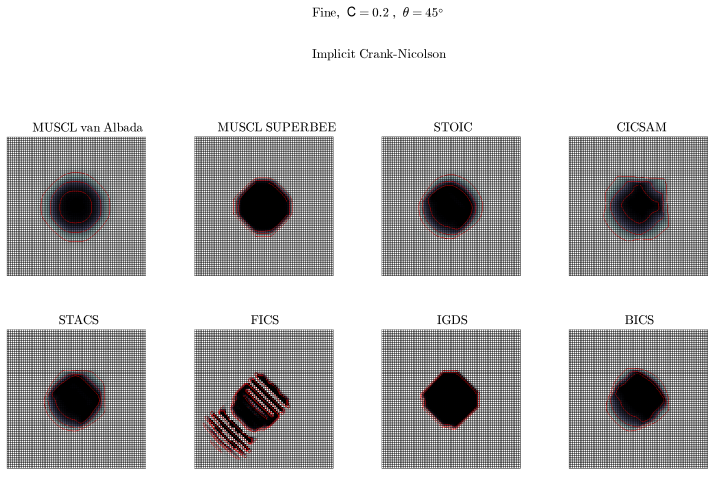


Figure A.27: Results from the circle case with $\mathsf{C} = 0.2$ and $\theta = 45°$, fine grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

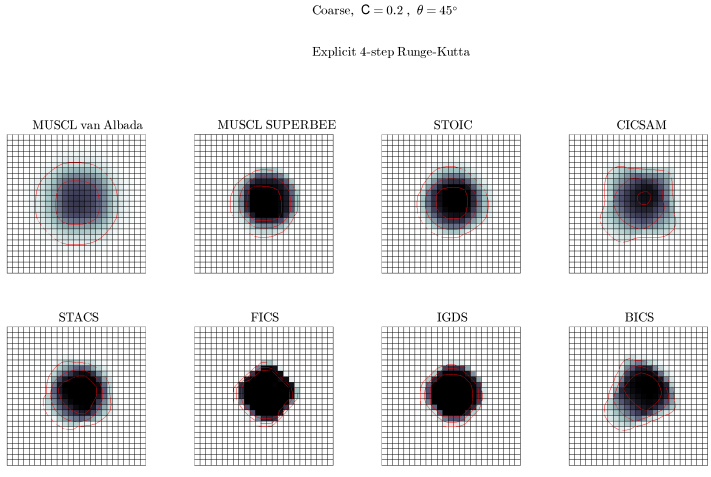


Figure A.28: Results from the circle case with $\mathsf{C} = 0.2$ and $\theta = 45°$, coarse grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

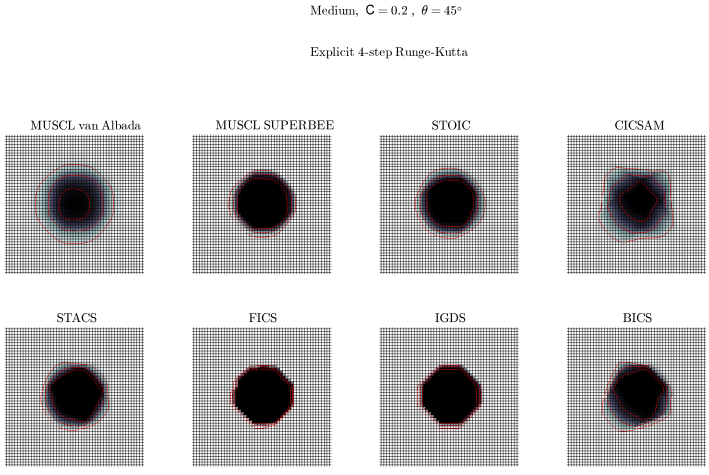


Figure A.29: Results from the circle case with $\mathsf{C} = 0.2$ and $\theta = 45°$, medium grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

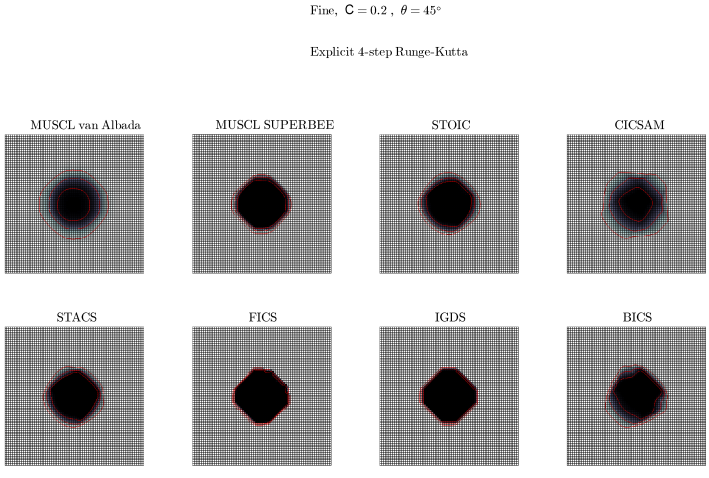


Figure A.30: Results from the circle case with $\mathsf{C} = 0.2$ and $\theta = 45°$, fine grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

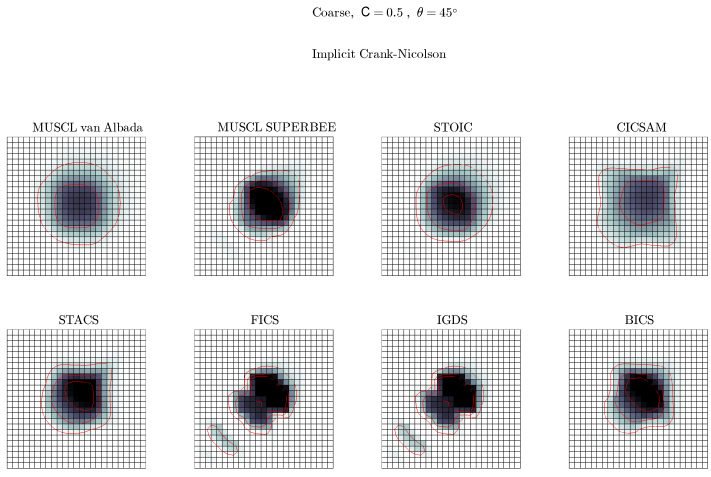


Figure A.31: Results from the circle case with $\mathsf{C} = 0.5$ and $\theta = 45°$, coarse grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

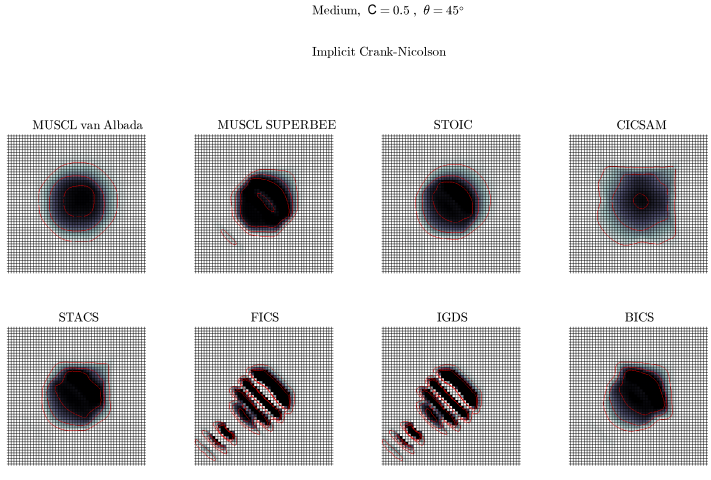


Figure A.32: Results from the circle case with $\mathsf{C} = 0.5$ and $\theta = 45°$, medium grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

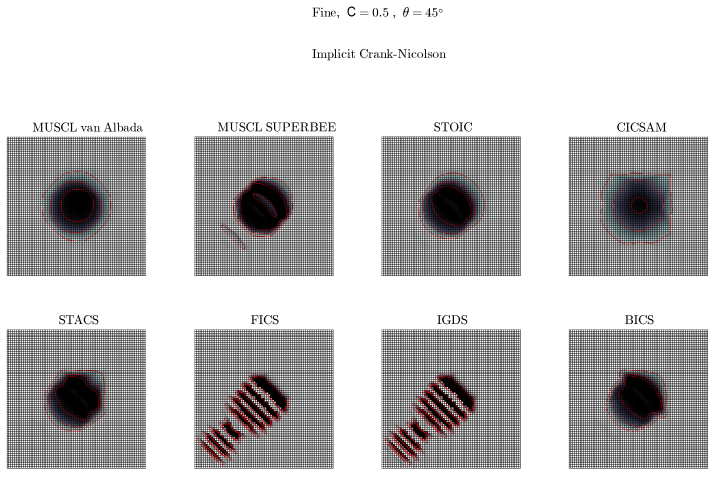


Figure A.33: Results from the circle case with $\mathsf{C} = 0.5$ and $\theta = 45°$, fine grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

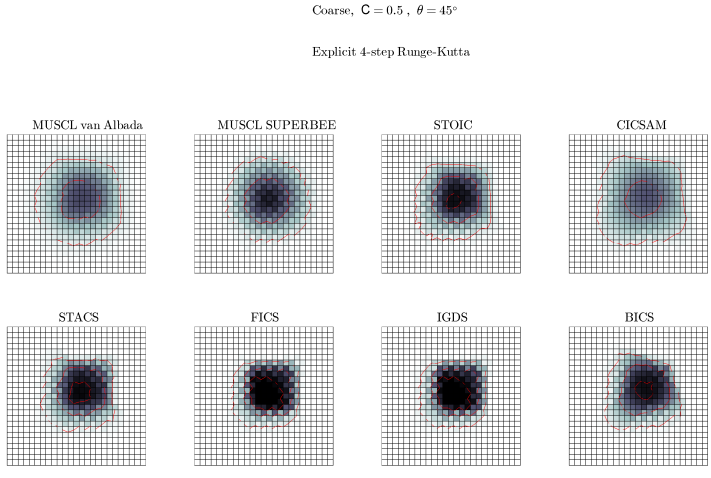


Figure A.34: Results from the circle case with $\mathsf{C} = 0.5$ and $\theta = 45°$, coarse grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

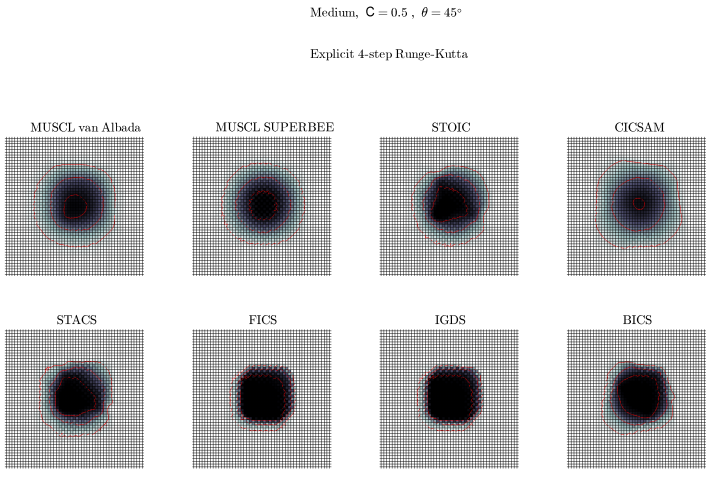


Figure A.35: Results from the circle case with $\mathsf{C} = 0.5$ and $\theta = 45°$, medium grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

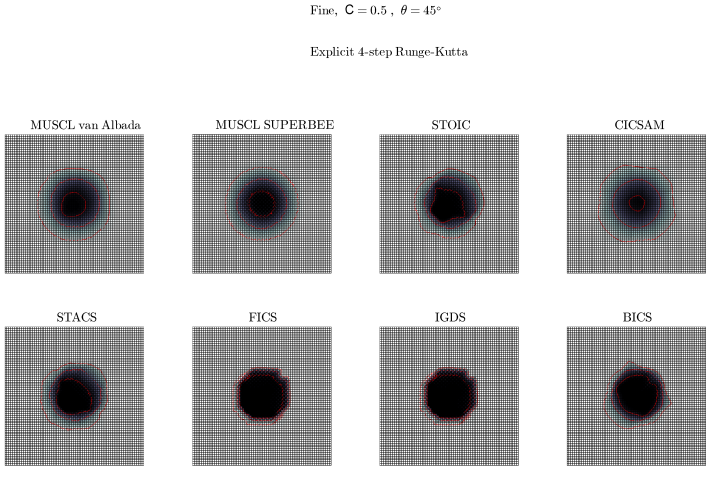


Figure A.36: Results from the circle case with $\mathsf{C} = 0.5$ and $\theta = 45°$, fine grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

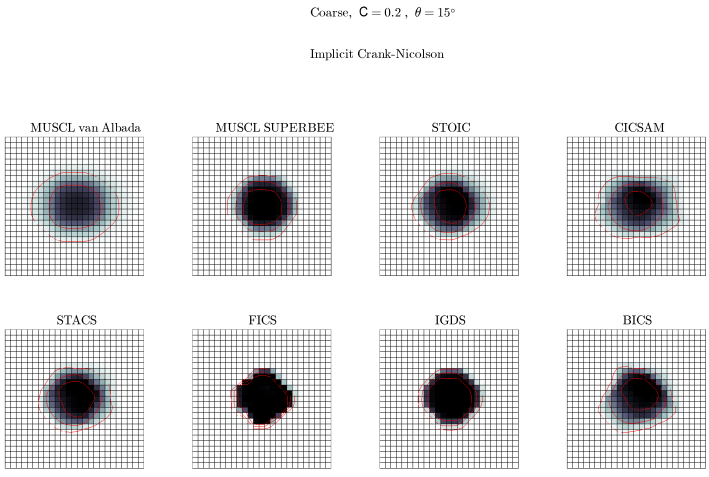


Figure A.37: Results from the circle case with $\mathsf{C} = 0.2$ and $\theta = 15°$, coarse grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

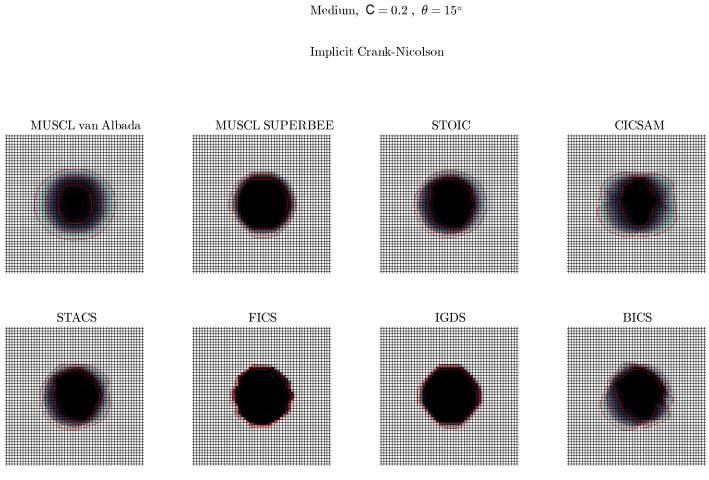


Figure A.38: Results from the circle case with $\mathsf{C} = 0.2$ and $\theta = 15°$, medium grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

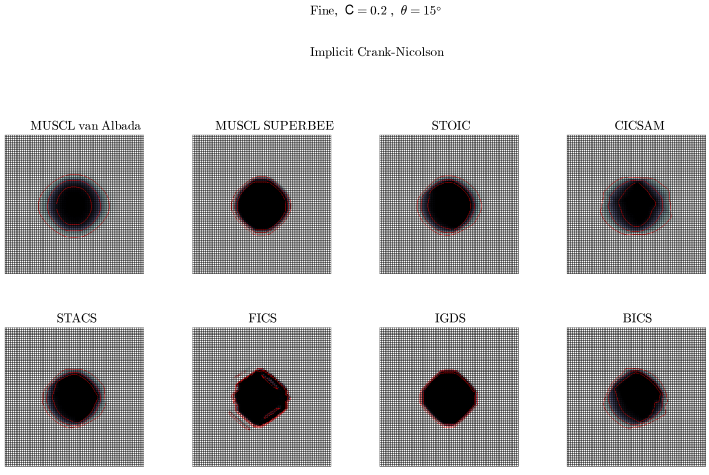


Figure A.39: Results from the circle case with $\mathsf{C} = 0.2$ and $\theta = 15°$, fine grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

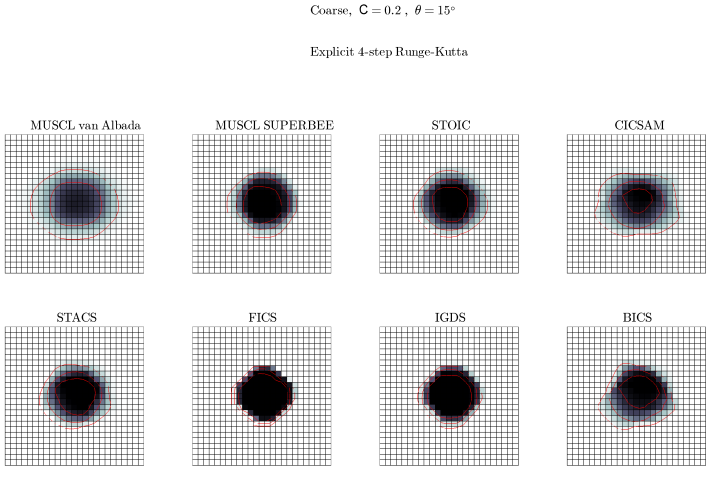


Figure A.40: Results from the circle case with $\mathsf{C} = 0.2$ and $\theta = 15°$, coarse grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

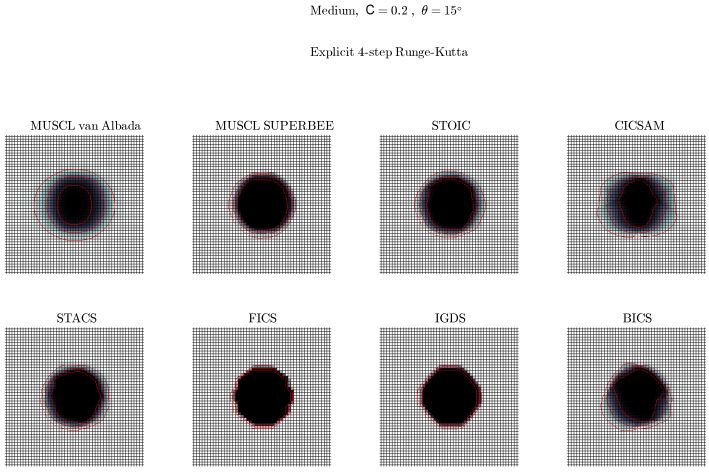


Figure A.41: Results from the circle case with $\mathsf{C} = 0.2$ and $\theta = 15°$, medium grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

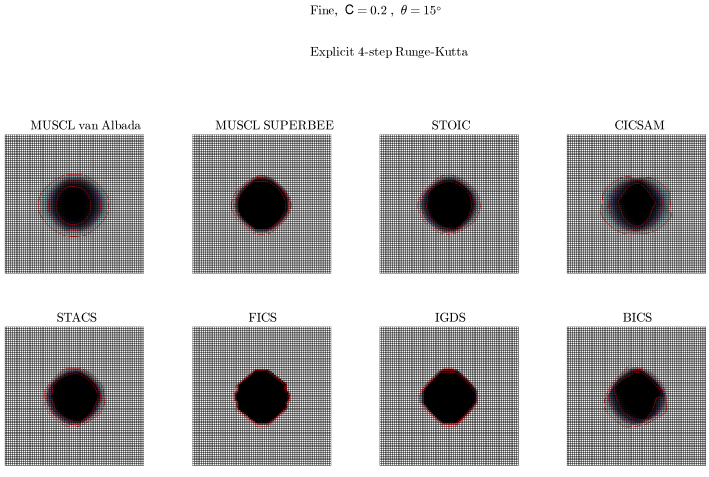


Figure A.42: Results from the circle case with $\mathsf{C} = 0.2$ and $\theta = 15°$, fine grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

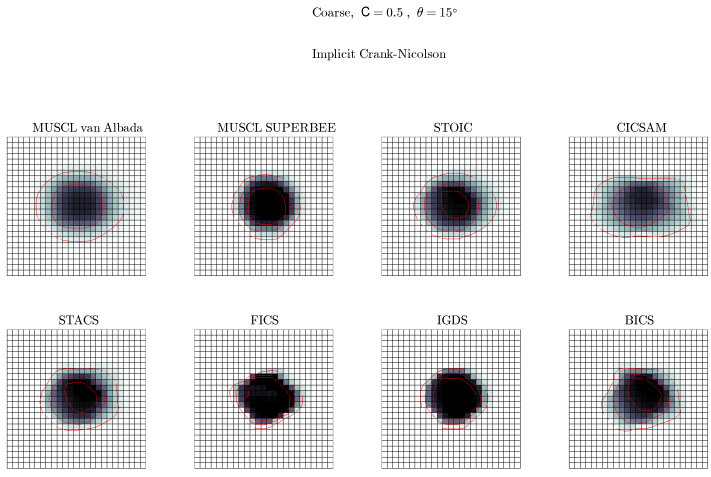


Figure A.43: Results from the circle case with $\mathsf{C} = 0.5$ and $\theta = 15°$, coarse grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

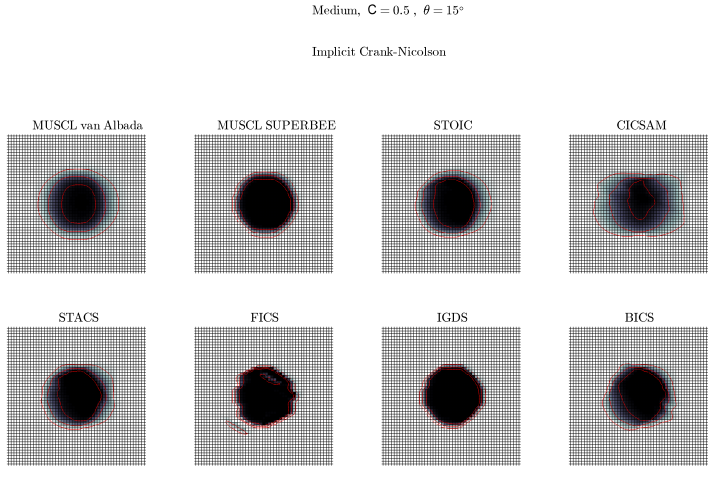


Figure A.44: Results from the circle case with $\mathsf{C} = 0.5$ and $\theta = 15°$, medium grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

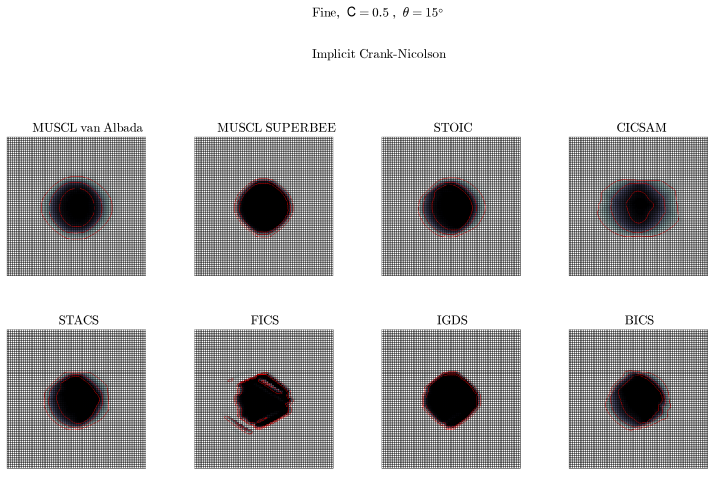


Figure A.45: Results from the circle case with $\mathsf{C} = 0.5$ and $\theta = 15°$, fine grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

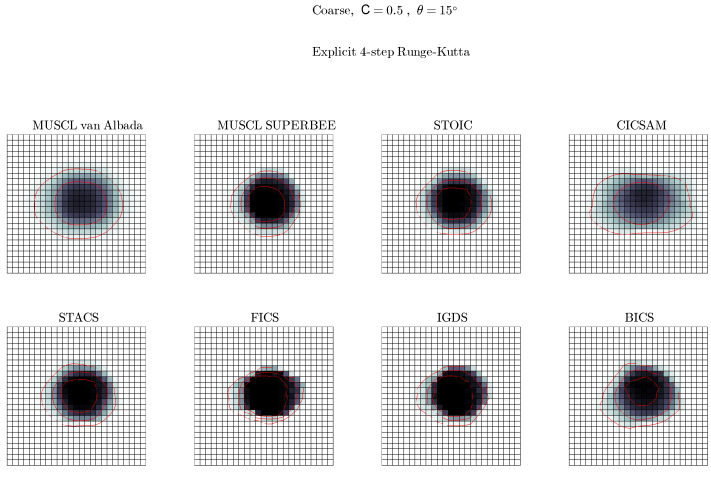


Figure A.46: Results from the circle case with $\mathsf{C} = 0.5$ and $\theta = 15°$, coarse grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

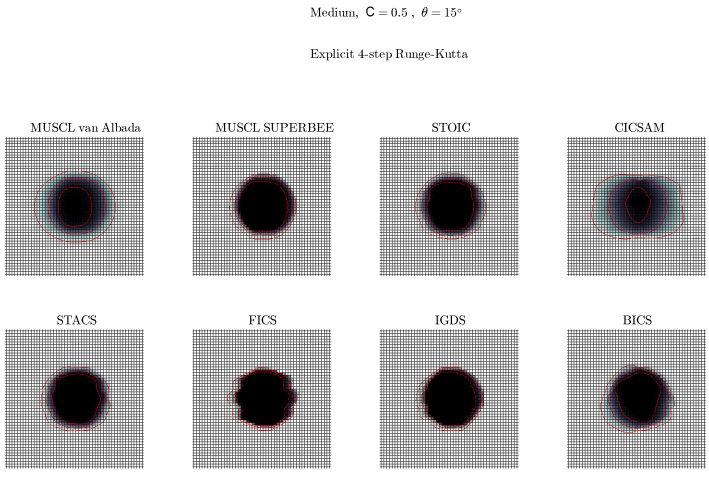


Figure A.47: Results from the circle case with $\mathsf{C} = 0.5$ and $\theta = 15°$, medium grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

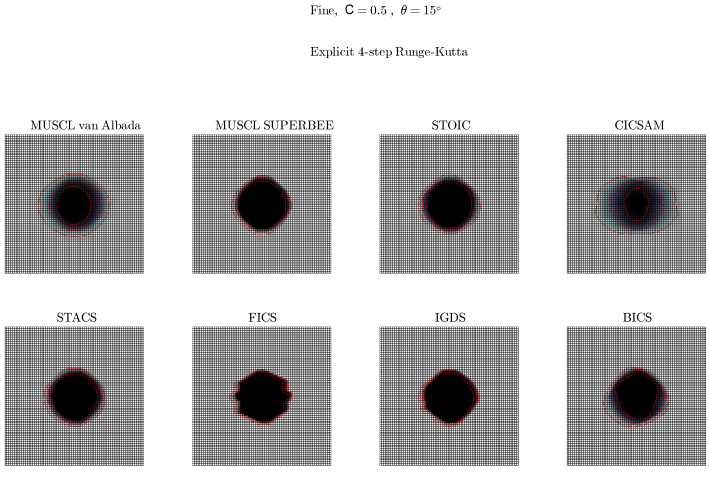
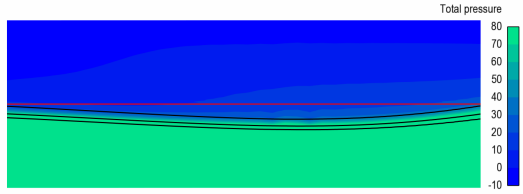


Figure A.48: Results from the circle case with $\mathsf{C} = 0.5$ and $\theta = 15°$, fine grid. Red lines are contours of $\alpha = [0.1, 0.5, 0.9]$.

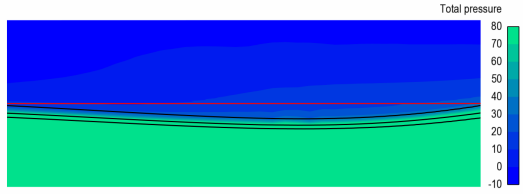
Appendix B

Results from the bump case

B.1 FINFLO, coarse grid

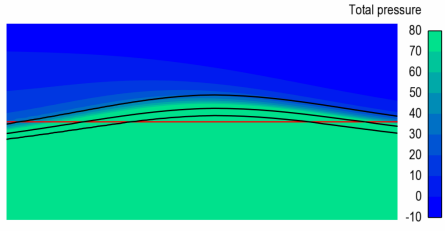


(a) Van Albada limiter

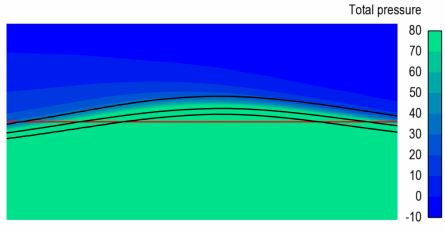


(b) SUPERBEE limiter

Figure B.1: First wave trough. The three thick black lines denote the void fraction $\alpha = [0.1, 0.5, 0.9]$, and the red line the still water level $z = 0$. The horizontal coordinate ranges as $x = [-0.5, 1.2]$ m and the vertical $z = [-0.3, 0.3]$ m

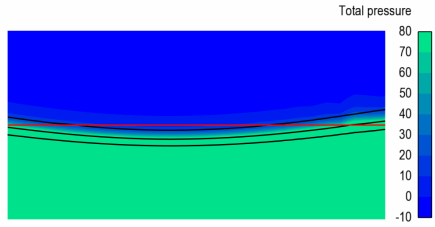


(a) Van Albada limiter

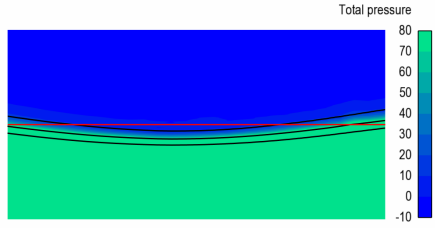


(b) SUPERBEE limiter

Figure B.2: First wave crest. The three thick black lines denote the void fraction $\alpha = [0.1, 0.5, 0.9]$, and the red line the still water level $z = 0$. The horizontal coordinate ranges as $x = [1.2, 2.4]$ m and the vertical $z = [-0.3, 0.3]$ m.

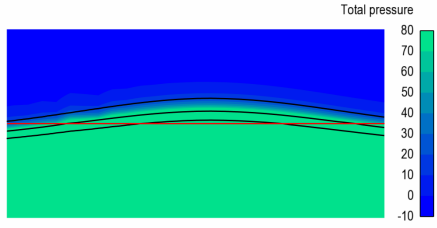


(a) Van Albada limiter

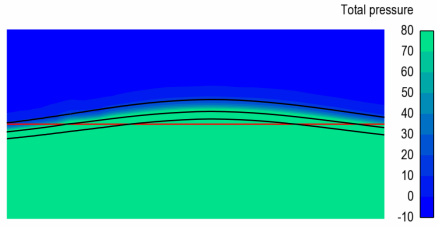


(b) SUPERBEE limiter

Figure B.3: Third wave trough. The three thick black lines denote the void fraction $\alpha = [0.1, 0.5, 0.9]$, and the red line the still water level $z = 0$. The horizontal coordinate ranges as $x = [4.4, 5.6]$ m and the vertical $z = [-0.3, 0.3]$ m.



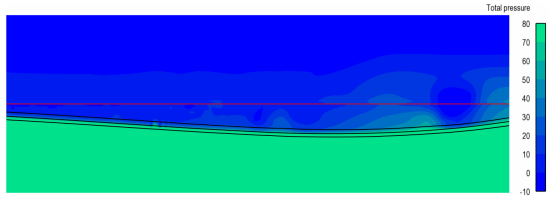
(a) Van Albada limiter



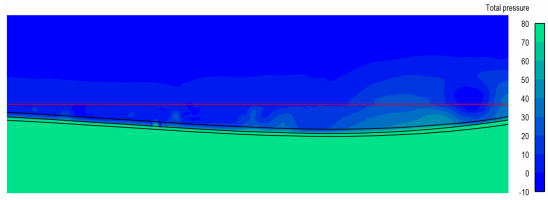
(b) SUPERBEE limiter

Figure B.4: Third wave crest. The three thick black lines denote the void fraction $\alpha = [0.1, 0.5, 0.9]$, and the red line the still water level $z = 0$. The horizontal coordinate ranges as $x = [5.3, 6.5]$ m and the vertical $z = [-0.3, 0.3]$ m.

B.2 FINFLO, fine grid

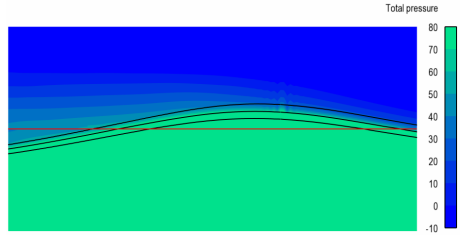


(a) Van Albada limiter

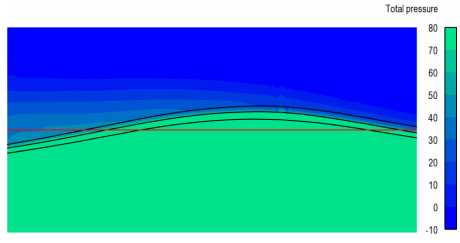


(b) SUPERBEE limiter

Figure B.5: First wave trough. The three thick black lines denote the void fraction $\alpha = [0.1, 0.5, 0.9]$, and the red line the still water level $z = 0$. The horizontal coordinate ranges as $x = [-0.5, 1.2]$ m and the vertical $z = [-0.3, 0.3]$ m

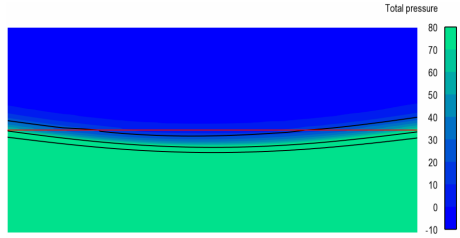


(a) Van Albada limiter

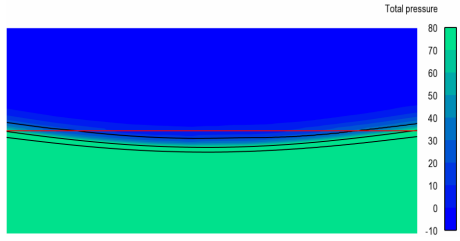


(b) SUPERBEE limiter

Figure B.6: First wave crest. The three thick black lines denote the void fraction $\alpha = [0.1, 0.5, 0.9]$, and the red line the still water level $z = 0$. The horizontal coordinate ranges as $x = [1.2, 2.4]$ m and the vertical $z = [-0.3, 0.3]$ m.

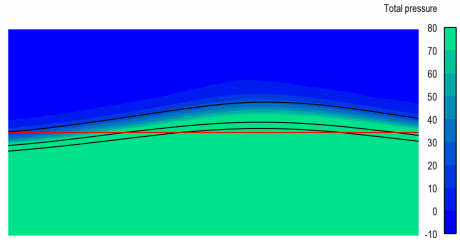


(a) Van Albada limiter

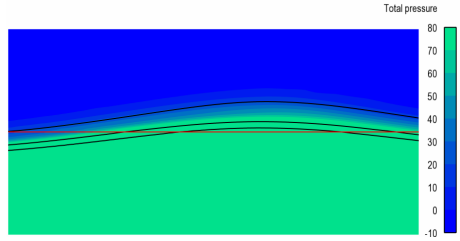


(b) SUPERBEE limiter

Figure B.7: Third wave trough. The three thick black lines denote the void fraction $\alpha = [0.1, 0.5, 0.9]$, and the red line the still water level $z = 0$. The horizontal coordinate ranges as $x = [4.4, 5.6]$ m and the vertical $z = [-0.3, 0.3]$ m.



(a) Van Albada limiter



(b) SUPERBEE limiter

Figure B.8: Third wave crest. The three thick black lines denote the void fraction $\alpha = [0.1, 0.5, 0.9]$, and the red line the still water level $z = 0$. The horizontal coordinate ranges as $x = [5.3, 6.5]$ m and the vertical $z = [-0.3, 0.3]$ m.

B.3 YAFFA

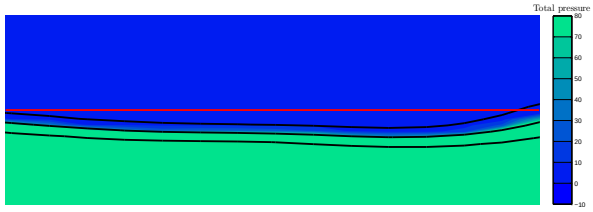


Figure B.9: First wave trough. The three thick black lines denote the void fraction $\alpha = [0.1, 0.5, 0.9]$, and the red line the still water level $z = 0$. The horizontal coordinate ranges as $x = [-0.5, 1.2]$ m and the vertical $z = [-0.3, 0.3]$ m.

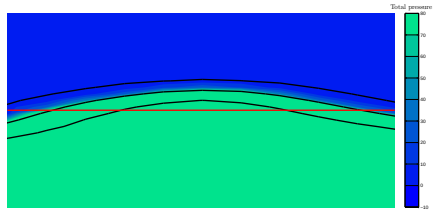


Figure B.10: First wave crest. The three thick black lines denote the void fraction $\alpha = [0.1, 0.5, 0.9]$, and the red line the still water level $z = 0$. The horizontal coordinate ranges as $x = [1.2, 2.4]$ m and the vertical $z = [-0.3, 0.3]$ m.

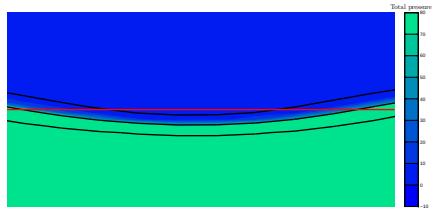


Figure B.11: Third wave trough. The three thick black lines denote the void fraction $\alpha = [0.1, 0.5, 0.9]$, and the red line the still water level $z = 0$. The horizontal coordinate ranges as $x = [4.4, 5.6]$ m and the vertical $z = [-0.3, 0.3]$ m.

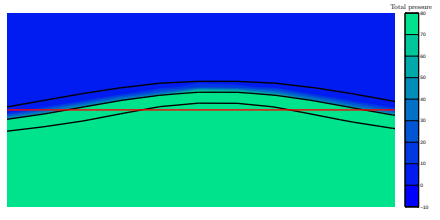


Figure B.12: Third wave crest. The three thick black lines denote the void fraction $\alpha = [0.1, 0.5, 0.9]$, and the red line the still water level $z = 0$. The horizontal coordinate ranges as $x = [5.3, 6.5]$ m and the vertical $z = [-0.3, 0.3]$ m.

Series title and number

VTT Science 98

Title	**Verification of a homogeneous mixture model for the free surface problem**
Author(s)	Ville Viitanen
Abstract	In this thesis, the applicability of the homogeneous mixture model of FINFLO for the free surface problem is studied. The free surface problem is fundamental in marine hydrodynamics, and a special case in two-phase flows. The work explores the basis of this type of modelling from mathematical and numerical viewpoints, and verifies the mixture model for the problem. The mathematical background of the problem is presented, together with the nature of it from the perspective of marine hydrodynamics. The bulk flow equations are usually averaged conditionally such that the governing equations of the multiphase model are formally the same as in the case of single-phase flow. It can be shown that one additional equation suffices for the description of the segregated phases. Here, the convection equation of the void fraction is utilized. The void fraction equation is derived in conservative form based on the incompressibility constraint of the individual phases. The convection of the void fraction corresponds to the so-called Riemann problem. This is studied thoroughly by developing a two-dimensional solver for the comparison of some well-known schemes for the spatial discretization of the convective quantity. This solver is applied to the convection of a discontinuous distribution of the void fraction. In addition, the so-called SUPERBEE limiter is implemented to the FINFLO code for the extrapolation of the convective void fraction. The numerical solution of the Navier-Stokes equations for simulations of two-phase flows is covered comprehensively. The code YAFFA, developed at the Aalto University, has a modern VOF model implemented, and for this reason, it is here used as a reference code. The solution algorithms, the computation of the convective quantities, the pressure correction stages as well as the treatment of the segregated phases in both of the codes are discussed in detail. The two-phase flow over a submerged ground elevation is computed using the codes FINFLO and YAFFA, and the forming free surface wave is compared to the corresponding results found from the literature. The aim of this thesis is to get acquainted with the nature of the problem in conjunction with the specific methodology used to solve such flows. This is done in order to understand the requirements and possible modifications needed for the model when we wish to accurately predict ship flow phenomena that are not solvable using the traditional free surface tracking strategies. This way, the verification of the mixture model of FINFLO is achieved.
ISBN, ISSN	ISBN 978-951-38-8315-7 (Soft back ed.) ISBN 978-951-38-8316-4 (URL: http://www.vttresearch.com/impact/publications) ISSN-L 2242-119X ISSN 2242-119X (Print) ISSN 2242-1203 (Online)
Date	January 2015
Language	English, Finnish abstract
Pages	192 p. + app. 35 p.
Name of the project	
Commissioned by	
Keywords	VOF, CFD, Two-phase flow, Hydrodynamics, Free surface, FINFLO, YAFFA, Numerical modelling, Convection, Interface capturing
Publisher	VTT Technical Research Centre of Finland Ltd P.O. Box 1000, FI-02044 VTT, Finland, Tel. 020 722 111

Julkaisun sarja ja numero

VTT Science 98

Nimeke	**Homogeenisen seosmallin verifiointi vapaan nestepinnan ongelmaan**
Tekijä(t)	Ville Viitanen
Tiivistelmä	Tässä työssä tutkitaan FINFLOn homogeenisen seosmallin soveltuvuutta vapaan nestepinnan ongelmaan. Vapaan nestepinnan ongelma on keskeinen laivahydrodynamiikassa, ja samalla monifaasivirtauksien erikoistapaus. Työssä perehdytään tällaisen mallinnuksen perusteisiin matemaattisessa ja numeerisessa mielessä, ja verifioidaan samalla seosmallia tälle ongelmalle. Työssä esitetään ongelman matemaattinen tausta sekä sen luonne laivahydrodynamiikan kannalta. Virtausta kuvaavat yhtälöt yleensä keskiarvostetaan ehdollisesti se. käytettävän monifaasimallin perusyhtälöt ovat muodollisesti samat, kuin yksifaasisessakin tapauksessa. Voidaan osoittaa, että tässä tapauksessa erillisten faasien kuvaukseen riittää yksi lisäyhtälö, joksi työssä otetaan aukko-osuuden konvektioyhtälö. Aukko-osuusyhtälö johdetaan säilymismuodossa perustuen faasien kokoonpuristumattomuusoletukseen. Mainittu lisäyhtälö vastaa luonteeltaan konvektioyhtälön ns. Riemann-probleemaa, ja tätä käsitellään perusteellisesti. Työssä kehitetään kaksidimensioinen ratkaisija, jolla vertaillaan tunnettuja menetelmiä konvektoituvan suureen paikkadiskretoinnille soveltamalla sitä epäjatkuvan aukko-osuusjakauman konvektioprobleemalle. Lisäksi implementoidaan FINFLOon ns. SUPERBEE-rajoitin konvektoituvan aukko-osuuden ekstrapolointiin. Työssä käsitellään kattavasti Navier-Stokes -yhtälöiden numeerista ratkaisua kaksifaasivirtausimulointimenetelmien kannalta. Referenssikoodiksi otetaan Aalto-yliopistossa kehitetty YAFFA, johon nykyaikainen VOF-malli on implementoitu. Muiden muassa koodien ratkaisualgoritmi, konvektoituvien suureiden laskenta, painekorjausvaihe sekä erottuneiden faasien käsittely kuvataan perusteellisesti. FINFLO- ja YAFFA -koodeilla lasketaan kaksifaasivirtaus vedenalaisen kummun yli, ja syntynyttä aaltokuviota verrataan myös kirjallisuudesta löytyviin tuloksiin. Työn ajatuksena on tutustua vapaan nestepinnan ongelman luonteeseen yhdessä tällaisen yleisemmän ratkaisutavan kanssa. Tavoitteena on ymmärtää mallille asetettavia vaatimuksia sekä sitä, millaisia modifikaatioita siihen tulisi tehdä, kun esim. pyritään ennustamaan tarkasti sellaisia laivavirtauksiin liittyviä ilmiöitä, joihin perinteiset pintaa seuraavat mallit eivät pysty. Tällä tavalla saatiin FINFLOn seosmallin verifiointi aikaiseksi.
ISBN, ISSN	ISBN 978-951-38-8315-7 (nid.) ISBN 978-951-38-8316-4 (URL: http://www.vtt.fi/julkaisut) ISSN-L 2242-119X ISSN 2242-119X (Painettu) ISSN 2242-1203 (Verkkojulkaisu)
Julkaisuaika	Tammikuu 2015
Kieli	Englanti, suomenkielinen tiivistelmä
Sivumäärä	192 s. + liitt. 35 s.
Projektin nimi	
Rahoittajat	
Avainsanat	VOF, CFD, kaksifaasivirtaus, hydrodynamiikka, vapaa nestepinta, FINFLO, YAFFA, numeerinen mallinnus, konvektio
Julkaisija	Teknologian tutkimuskeskus VTT Oy PL 1000, 02044 VTT, puh. 020 722 111

Verification of a homogeneous mixture model for the free surface problem

The free surface problem is fundamental in marine hydrodynamics, and a single most distinct feature distinguishing the field in the discipline of fluid dynamics. At the same time, the free surface problem is essentially a special case in two-phase flows.

This publication investigates the applicability of a two-phase model for the free surface problem. The nature of the problem is studied in conjunction with a specific CFD methodology capable to solve such flows. The current practices, solution methods and discretizations used in CFD are detailed comprehensively.

In this publication, a homogeneous mixture model that has been implemented in FINFLO, a general purpose CFD code, is verified for the free surface problem. The two-dimensional flow over a bump submerged in water is computed using both the mixture model of FINFLO, and a reference code that employs the standard VOF model. The resulting wave formation is also compared to the corresponding results from the literature. In addition, the special solution methods and discretizations for the VOF method are studied in detail by developing a two-dimensional solver for the convection problem.

A powerful free surface capturing model is practically the only option for hydrodynamic analyses in many areas of marine technology. For instance, the predictions of the free surface flow around complex geometries, of the slamming type wave loads, of the propeller ventilation or of the seaworthiness of the vessel cannot be done with CFD using the traditional free surface tracking approach.

ISBN 978-951-38-8315-7 (Soft back ed.)
ISBN 978-951-38-8316-4 (URL: http://www.vttresearch.com/impact/publications)
ISSN-L 2242-119X
ISSN 2242-119X (Print)
ISSN 2242-1203 (Online)