NORGES TEKNISK-NATURVITENSKAPELIGE
UNIVERSITET

Solution Methods for Incompressible Viscous Free Surface Flows: A Litterature Review

by

Samuel R. Ransau

PREPRINT
NUMERICS NO. 3/2002

NORWEGIAN UNIVERSITY OF SCIENCE AND
TECHNOLOGY
TRONDHEIM, NORWAY

This report has URL `http://www.math.ntnu.no/preprint/numerics/2002/N3-2002.ps`
Address: Department of Mathematical Sciences, Norwegian University of Science and
Technology, N-7491 Trondheim, Norway.

Solution Methods for Incompressible Viscous Free Surface Flows : A Litterature Review

Samuel R. Ransau

November 2002

Contents

Chapter 1

Introduction

1.1 Background and motivations

Free surface flow in the vicinity of rigid bodies is a very important class of problems for naval architects and marine engineers. The fluid flow around, for example, a full ship hull is nonlinear, turbulent and may involve physical phenomena such as breaking waves, and is therefore a very challenging problem. Maneuvring and resistance perfomances of ships may differ to a great extent by even small differences in the hull form. In restricted waters, knowledge and understanding of ship maneuvrability is very important. Therefore an accurate calculation of drag and lift forces of a ship is a major concern in the design of a ship. So far, experience and experimental data (mainly from model tests) are the main sources of information used in the design of ships. However, two problems are usually associated with the experimental approach: its cost and the scaling effects in model test data. Indeed, maintaining the Froude number similarity usually leads to a violation of the Reynolds number similarity and vice-versa. Thus experimental results from model tests and full scale tests are not similar. Consequently, other types of prediction methods have been seeked for.

Theoritical predictions of hydrodynamic characteristics are usually limited to simple geometries and physics. On the other hand, the computational tools currently available for ship design use approximations that simplify the equations representing the physical model. These approximations are usually referred to as potential flow approximations. Methods based on potential theory can predict wave resistance quite well. According to Beck et al. [5] this type of codes are ready to be used in industry for wave resistance and sea-keeping. These methods are very cost and time efficient and still receive much attention by the marine engineers and naval architects. However, several reviews (Gorski [25], Larsson et al. [49]) show the limitations of potential flow computations in reproducing some of the important physical features of the flow field around a ship hull. They state that potential flow based methods are not sufficient for many important applications and that viscous flow calculation methods need to be considered and developed. Viscous methods in ship hydrodynamics are based on the solution of the incompressible Navier-Stokes equations. The incompressibility approximation can be considered as a good one for ship hydrodynamics applications since, according to Tannehil et al. [81] fluid flows can be considered as incompressible for Mach numbers ($M = \frac{V}{a}$) smaller than 0.3. V is the fluid velocity and a the speed of sound in water. For the temperature range of interest in marine hydrodynamics applications, the speed of sound in water (tabulated in [93]) is about 1460 m/s. In order for the Mach number M to be smaller than 0.3, the fluid velocity must be smaller than 438 m/s (or equivalently 851 knots for a ship!!), which is always the case.

Moreover, the focus will be put on methods for three-dimensional problems since Barkley & Henderson [4] and Henderson [31] show that for high Reynolds number simulations, three-dimensional computations must be performed.

Nowadays, most of the existing codes computing three-dimensional incompressible viscous fluid flows are based on the Reynolds-Averaged Navier-Stokes (RANS) [83] formulation of the flow equations. This approach accounts for turbulence through modelling of the momentum effects. The increasing power of computers allows for more advanced methods like the Large Eddy Simulation (LES) technique where the large scale motions are computed directly while the small scale motions are modelled, or like the Direct Numerical Simulation (DNS) where all scales are computed. The latter technique is however currently possible only for problems with very low Reynolds number.

Another very important aspect in ship hydrodynamic applications is the prediction of the free surface location and evolution. It is a critical issue in naval applications. For instance, breaking of waves around a ship hull can increase considerably the ship's resistance. The signature of a ship, i.e. the wavy configuration and the turbulence pattern in the wake of the ship, relevant for ship detection, is also another example where an accurate prediction of the free surface is important. Free surface prediction is a very difficult problem which is still not completely solved. Many special techniques have been developed through the years in order to predict the motion of the flow around surface piercing bodies. Two main trends exist: the interface tracking type of methods and the interface capturing type of methods. Both types have their strengths and weaknesses.

1.2 Outline of the literature review

In the sequel, the governing equations of incompressible fluid flows and some of the solution methods used for solving them will be presented. Indeed while seeking for a numerical solution of these equations, the algebraic system arising from the spatial and temporal discretisation can be very large when dealing with real life applications. Moreover an extra-difficulty is added by the coupling between velocity and pressure fields. Various methods have been devised in order to reduce the computational complexity either by a decoupling of velocity and pressure or by the introduction of a small perturbation in the mass conservation equation.

Secondly, some of the most used methods, in ship hydrodynamic applications, for the prediction of free surface flows interacting with general curved boundaries will be presented, and some of their advantages and disadvantages will be enlighted.

Then some short comments concerning initial and boundary conditions will be made and finally, some interesting test cases that will be used for validation purposes will be described.

This text is not meant to be an exhaustive review of the existing methods for simulation of flow with free surface but only a presentation of some of the most commonly used methods in marine hydrodynamics.

Chapter 2

General solution methods

The cornerstone of Computational Fluid Dynamics (CFD) is the fundamental governing equations of fluid dynamics. These equations represent a mathematical model for describing viscous (Newtonian) flows. They are usually referred as Navier-Stokes equations and were first derived in 1822 by the French Engineer C. M. L. H. Navier upon the basis of a suitable molecular model. Curiously, these equations were recognized, by the scientist community of that time, to be totally inconsistent from a physical point of view for several materials, and in particular for liquids. It was only twenty-three years later, in 1845, that the english scientist G. H. Stokes derived the same equations in a quite general way, by means of the theory of continuum.
In this chapter, these equations will be shortly presented, as well as different solution methods for the three-dimensional incompressible bulk flow.

2.1 Governing equations

These Navier-Stokes equations are mathematical statements for the following three physical laws:

1. mass is conserved,

2. Newton's second law (**F**=m**a**),

3. energy is conserved.

In the case where the fluid is subject to the action of a external force $\mathbf{f}$, the general Navier-Stokes equations can be written as follows:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &= 0 && (2.1)\\ \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\,\mathbf{u} &= \frac{1}{\rho} \nabla \cdot \boldsymbol{\sigma} + \mathbf{f} && (2.2)\\ \frac{\partial (c_p \rho T)}{\partial t} + c_p \rho \mathbf{u} \cdot \nabla T &= \varepsilon + \nabla \cdot (\lambda \nabla T) && (2.3) \end{aligned}$$

where $\boldsymbol{\sigma}$ is the stress tensor given by:

$$\sigma_{ij} = -p\delta_{ij} + 2\mu S_{ij} \qquad (2.4)$$

p is the pressure, S the rate-of-strain tensor and μ the dynamic viscosity. Further, $\mathbf{u}$ is the fluid velocity, ρ the fluid density, T the temperature, ν the kinematic viscosity, c_p the heat capacity, λ the heat conductivity and ε the viscous dissipation. $\mathbf{f}$ and ε are considered to be known.

We will in the following make two main assumptions:

- the fluid is incompressible and homogeneous: the density is then constant in time and space,
- the heat conductivity and capacity are constant.

The first assumption allows us to write equation of mass (2.1) as

$$\nabla \cdot \mathbf{u} = 0 \qquad (2.5)$$

While using both assumptions, the energy equation (2.3) becomes:

$$c_p \rho \frac{\partial T}{\partial t} + c_p \rho \mathbf{u} \cdot \nabla T = \varepsilon + \lambda \Delta T \qquad (2.6)$$

If further the viscosity is taken to be constant throughout the fluid, the momentum equation (2.2) can then be written as:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\,\mathbf{u} = -\nabla p + \nu \Delta \mathbf{u} + \mathbf{f} \qquad (2.7)$$

Here p represents the kinematic pressure (i.e. pressure divided by the density). Equations (2.5), (2.7) and (2.6) are a mixed set of elliptic-parabolic equations where the unknowns are the velocity $\mathbf{u}$, the kinematic pressure p and the temperature T.
Note that the energy equation (2.6) can be uncoupled from the other two ones since it is the only equation where the temperature appears explicitly.

However, the fluid properties c_p, λ and the kinematic viscosity ν usually depend on the temperature. Therefore this uncoupling is exact if these fluid properties (and in particular the viscosity) are independent of temperature. For many applications, the temperature changes are either insignificant or unimportant, and it is not necessary to solve the energy equation. However, if one wishes to find the temperature distribution, this can easily be done, since the unsteady energy equation is parabolic (provided that the velocity field $\mathbf{u}$ is known). With this in mind, we will here only focus on the solution of the continuity and momentum equations (2.5) and (2.7).

Equations (2.2) is said to be the stress formulation of the momentum equation, while equation (2.7) is the Laplace formulation. The latter form is simpler to implement and work with. However, there might be cases where the stress formulation is preferred. This is for example the case when using turbulence models or if a force is given on a boundary. Nevertheless, equations (2.5) and (2.7) are the governing equations (for incompressible viscous fluid flows) that we will deal with through this section, unless something else is explicitly stated.
In addition to these equations, appropriate boundary and initial conditions for the flow variables are needed (see chapter 4) in order to close the system and have a well-posed problem.

2.2 Methods for the bulk flow equations

The analytical solution of problem (2.7), (2.5), together with appropriate initial and boundary conditions is seldom known in a closed-form. Therefore, one needs to seek for a numerical solution. In order to compute this solution, one needs to discretise the equations (as well as the initial and boundary conditions) both in time and space. Although the approaches (choice of solution method and discretisation technique) can be very different, the combined time and spatial discretisation reduces the problem to finding, at each time step, the solution of an equation system of the form:

$$A\mathbf{y}^{n+1} = \mathbf{b}^{n+1} \tag{2.8}$$

where A is a (known) coefficient-matrix, $\mathbf{b}^{n+1}$ is a known vector and $\mathbf{y}^{n+1}$ the unknown vector.
Several solution methods for solving the governing equations of an incompressible viscous fluid flow, and arrive at a system of the form (2.8) have been developed and used successfully over the years. These methods can be classified into two main types:

- methods solving for the so-called primitive variables i.e. the velocity and the pressure,

- methods solving for the derived (or non-primitive) variables which are the vorticity and stream function.

Each type can be further subdivided.

2.2.1 Vorticity-stream function approach

Methods solving for the *derived variables*, also referred as *vorticity-stream function approaches*, have been very popular when it comes to solving the two-dimensional incompressible Navier-Stokes equations. In this type of approach, a change of variables that replaces the velocity components with the vorticity ω and the stream function ψ is made. The vorticity vector is defined as:

$$\boldsymbol{\omega} = \nabla \times \mathbf{u} \tag{2.9}$$

while the scalar value of the vorticity ω is defined as the z-component of the vorticity vector:

$$\omega = \nabla \times \mathbf{u} \cdot \mathbf{k} \tag{2.10}$$

where $\mathbf{k}$ is the unit vector in z-direction.
If the computational domain Ω is a simple-connected domain of $I\!R^2$, a well-known result (see [69]) states that a vector function $\mathbf{u} \in (\mathbf{L^2(\Omega)})^{\mathbf{2}}$ satisfies $\nabla \cdot \mathbf{u} = \mathbf{0}$, if and only if there exists a function $\psi \in H^1(\Omega)$, called *stream function*, such that in a Cartesian coordinate system one has:

$$\frac{\partial \psi}{\partial x} = -v \quad and \quad \frac{\partial \psi}{\partial y} = u \tag{2.11}$$

In the two-dimensional case, taking the curl of the momentum equation (2.7), using the continuity equation (2.5) and inserting the previously defined scalar vorticity and stream function gives:

$$\frac{\partial \omega}{\partial t} + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = \nu\left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2}\right) \tag{2.12}$$

or

$$\frac{D\omega}{Dt} = \nu \Delta \omega \tag{2.13}$$

where $\dfrac{D()}{Dt} = \dfrac{\partial()}{\partial t} + \mathbf{V} \cdot \nabla()$ is the substantial derivative.
Equation (2.12), or equivalently (2.13), is a parabolic partial differential equation often referred as *vorticity transport equation*. The one-dimensional version of this equation is the advection-diffusion equation which is often used as an equation model:

$$\frac{\partial \omega}{\partial t} + u\frac{\partial \omega}{\partial x} = \nu\frac{\partial^2 \omega}{\partial x^2} \tag{2.14}$$

and therefore many successfull solution methods are available.
An equation for the stream function ψ can also be obtained by combining equations (2.10) and (2.11):

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega \tag{2.15}$$

or again

$$\Delta\psi = -\omega \tag{2.16}$$

This is the elliptic Poisson equation.

As a result of this change of variables, we have been able to, first of all eliminate the pressure (which is usually a source of problems) from the set of equations and secondly to separate the mixed elliptic-parabolic incompressible Navier-Stokes equations into one parabolic equation and one elliptic equation. These equations are usually solved sequentially using the following time-marching procedure:

1. Specify initial values for ω and ψ,
2. Solve the vorticity transport equation for ω at each interior grid point at time t_{n+1},
3. Solve the Poisson equation for ψ at all points using the new values of ω at the interior points,
4. Find the velocity components u and v by applying equations (2.11),
5. Determine the values of ω on the boundary using the values of ω and ψ at interior points,
6. Return to step 2 if the solution is not converged.

The solution of the vorticity-stream function system can also be computed using techniques of coupled type rather than sequential type. Rubin and Khosla [73] solved the 2×2 coupled system for ω and ψ efficiently by using a strongly implicit procedure.

Here, the implementation of the boundary conditions requires special attention. For solid boundaries where the no-slip condition is to be imposed, equations (2.11) gives:

$$\psi = 0 \quad \textit{and} \quad \frac{\partial\psi}{\partial n} = 0 \tag{2.17}$$

on the solid surface. The first condition is used as it is with the Poisson equation for the stream-function ψ, while the second one is used in the construction of a boundary condition for the vorticity. See [23] for more details and for the specification of inflow and outflow boundary conditions.

Note that the pressure does not appear explicitly in this approach. However when it is of interest, it can be computed without difficulty by simply solving a Poisson equation obtained by taking the divergence of the momentum equation (2.7) (cf. [23], [81]):

$$\Delta p = 2\left(\frac{\partial u}{\partial x}\frac{\partial v}{\partial y} - \frac{\partial v}{\partial x}\frac{\partial u}{\partial y}\right) \tag{2.18}$$

Here p is the kinematic pressure (i.e. the pressure divided by the density).

Hence, in the two-dimensional case, inserting $\boldsymbol{\omega}$ and ψ in equations (2.5) and (2.7) yields to a Poisson equation for the stream function and a transport equation for the vorticity to solve. This approach is very attractive since it reduces the number of equations to solve and does not require any pressure computations. However, it loses of its attractiveness when it comes to three-dimensional applications. Indeed, the extension of the vorticity-stream function approach to three-dimensional problems is not trivial since stream functions do not exist for truly three-dimensional problems. However, it is possible to generalize this appraoch by introducing a *vector potential* [3], [23], [81] to obtain a *vorticity-potential formulation*. It is also possible to derive a *vorticity-velocity* fomulation as shown in [23]. In either case, the required computational effort is considerably increased since now 3 Poisson equations (for the vector potential or for the velocity components) and 3 transport equations for the vorticity need to be solved. Moreover, the boundary conditions become very complex [34]. As a consequence, the incompressible Navier-Stokes equations are most often solved in their primitive-variable form (u, v, w, p) for three-dimensional problems. Even for two-dimensional problems, the use of primitive variables is quite common.

2.2.2 Primitive variable approach

Methods for solving Navier-Stokes equations in their primitive-variable form have been, for a long time, and still are an area of intense research. Methods for solving this form of the Navier-Stokes equations can be divided into two main groups (each of them can further be subdivided): the *coupled*-type and *uncoupled*-type of methods.

2.2.2.1 Approaches of coupled-type

There are two main trends in the coupled-type of approach: the direct methods and the pseudo-compressibility methods.

Direct methods

The first one is to obtain a coupled system of equations from direct discretisation of the governing equations (i.e. momentum equation, continuity equation, boundary conditions and eventually an equation for the free surface). The resulting global system is usually difficult to solve and standard methods can usually not be used. Nonlinear iteration techniques (e.g. Newton-Raphson) or *predictor-corrector* techniques must be used (see [87] and the references therein). This *direct solution method* of the Navier-Stokes equations is usually computationally very expensive, especially for three-dimensional applications when serial programming is performed. For parallel computations this method can be attractive if the number of processors used is large enough so that a direct solution of the algebraic system is possible on each processor.

Pseudo-incompressibility methods

The second type of *coupled* solution methods takes advantages of techniques developed primarily for compressible flows: the *artificial compressibility method* introduced by Chorin [12], the *penalty method* of Hughes et al. [39] or the Petrov-Galerkin elliptic pressure regularisation [69]. They are usually known as *pseudo-compressibility* (or *quasi-compressibility*) methods.

Artificial compressibility method

In the *artificial compressibility method*, the continuity equation is modified to include an artificial compressibility term of the form $\dfrac{\partial \rho^*}{\partial t^*}$ that vanishes when the steady-state solution is reached. This new term is sometimes rewritten as $\epsilon \dfrac{\partial p}{\partial t^*}$ using the equation of state $p = \dfrac{\rho^*}{\epsilon}$ where ϵ is the artificial compressibility factor, ρ^* the artificial density and t^* a fictitious time that is analoguous to real time in compressible flow. The artificial compressibility parameter can be seen as a device to generate a well-posed system. This new term gives to the resulting Navier-Stokes equations a mixed hyperbolic-parabolic character which can be solved using a standard time-dependent approach after substituting t with t^* in the momentum equation. One finally obtains the following problem:

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t^*} + (\mathbf{u} \cdot \nabla)\,\mathbf{u} = -\nabla p + \nu \Delta \mathbf{u} + \mathbf{f} \\ \nabla \cdot \mathbf{u} + \epsilon \dfrac{\partial p}{\partial t^*} = 0, \qquad p(t^* = 0) = p_0 \end{cases} \tag{2.19}$$

This method is however only applicable to steady-flow problems because it is not time accurate.

Penalty method

In the *penalty method*, the continuity equation is also modified but this time, the added term is of the form ϵp where ϵ is a (small) penalty parameter such that $\nabla \cdot \mathbf{u} \longrightarrow 0$ as $\epsilon \longrightarrow 0$ if the pressure p is finite:

$$\nabla \cdot \mathbf{u} + \epsilon p = 0 \tag{2.20}$$

Inserting this modified continuity equation into the momentum equation yields to a *penalised momentum equation* which does not contain the pressure and can then be solved directly for the velocity:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\,\mathbf{u} = \frac{\mathbf{1}}{\epsilon} \nabla\,(\nabla \cdot \mathbf{u}) + \nu \boldsymbol{\Delta} \mathbf{u} \tag{2.21}$$

The pressure can thereafter be found by $p = -\dfrac{1}{\epsilon} \nabla \cdot \mathbf{u}$.

Unlike the artificial compressibility method, the penalty method can also be used for unsteady problems since it is time accurate. It appears then as a significant simplification

compared to the **u**-p system and this is one of the reasons for its popularity. However it has an important disadvantage: it is not, a priori, obvious how to choose the penalty parameter ϵ. If ϵ is too large, the continuity equation will be poorly approximated. If ϵ is too small, one may get numerically $\epsilon p = 0$ and the penalised momentum equation (2.21) simply reduces to $\nabla (\nabla \cdot \mathbf{u}) \simeq 0$.
Even though the pressure can be computed from equation (2.20), once the velocity field is known, it is of interest to derive an analog of the pressure Poisson equation when the penalty method is used [28]. The possible large effects of the small penalty parameter ϵ on the pressure can in that way be prevented. It is particularly important to derive this analog pressure Poisson equation for unsteady problems where transient phenomena are of importance since the undesirable effects of ϵ on the pressure happen only for small time. Those effects are related to a "spurious transient penalty shock wave" [28].

Petrov-Galerkin pressure regularisation
The Petrov-Galerkin regularisation, was first introduced by Hughes et al. in 1986 [40] in the context of finite element method, in order to circumvent the LBB-condition and enhance the stability properties of numerical schemes for Stokes and Navier-Stokes problems. Here the pressure is added in the form of its laplacian $-\epsilon\Delta p$ leading to the following modified continuity equation:

$$\nabla \cdot \mathbf{u} - \epsilon\Delta p = 0, \quad \nabla p \cdot \mathbf{n}_{|\Gamma} = 0 \tag{2.22}$$

All the previous three methods have the effect of relaxing the incompressibility constraint. The perturbation parameter ϵ must be sufficiently large in order to produce a significant regularisation effect (and act on the stability) but as small as possible to minimize the perturbation in the continuity equation.

2.2.2.2 Approaches of uncoupled type

The second group of general solution techniques, for the primitive-variable form of the Navier-Stokes equations, contains techniques of *uncoupled* type. These solutions methods are very popular techniques which separate the computation of the velocity from the computation of the pressure. The idea is to solve sequentially a number of smaller, linear equation systems instead of a larger, usually nonlinear and slowly converging one. Here, the momentum equation is solved for the velocity components (using an estimate of the pressure [90] or not [45]), then a Poisson equation for the pressure is solved and finally the velocity field is updated by using the newly found pressure field. The third step, usually called a correction step, is necessary in order to insure an incompressible velocity field. Indeed, the first step does not usually yield to a velocity field satisfying the continuity equation. In this type of methods, the incompressibility constraint (continuity equation) does not appear directly, but is accounted for through the Poisson equation solved for the pressure. This type of techniques are known under several names: *projection method, fractional-step method* or *pressure-correction method*, and was first introduced by Chorin [13] and Temam [82] and have been very studied since (see for example [26], [29]).

The projection method

The theoritical background for this method is the decompisiton theorem of Ladyzhenskaya [68] allowing to write any function in $L^2(\Omega)$ as the sum of a divergence-free part and a curl-free part.

The semi-discrete (discretised in time but not in space) form the projection method can be written as:

1. Find an intermediate velocity $\tilde{\mathbf{u}}^{n+1}$ in the domain Ω as the solution of:

$$\frac{\tilde{\mathbf{u}}^{n+1}-\mathbf{u}^n}{\delta t} = -\left(\mathbf{u}^* \cdot \nabla\right)\mathbf{u}^{**} + \nu\Delta\tilde{\mathbf{u}}^{n+1} + \mathbf{f}^{n+1} \tag{2.23}$$

   with $\tilde{\mathbf{u}}^{n+1}$ satisfying the boundary condition on $\partial\Omega$. $\mathbf{u}^*$ and $\mathbf{u}^{**}$ are to be chosen suitably for the treatment of the nonlinear term (e.g. $\mathbf{u}^* = \mathbf{u}^{**} = \mathbf{u}^n$ for the explicit Euler, $\mathbf{u}^* = \mathbf{u}^{**} = \mathbf{u}^{n+1}$ for the implicit Euler and $\mathbf{u}^* = \mathbf{u}^n$, $\mathbf{u}^{**} = \mathbf{u}^{n+1}$ for the Euler semi-implicit).

2. Determine the pressure p^{n+1} and the end-of-step velocity $\mathbf{u}^{n+1}$ as the solution of the problem:

$$\left\{\begin{array}{lcl} \dfrac{\mathbf{u}^{n+1}-\tilde{\mathbf{u}}^{n+1}}{\delta t} & = & -\nabla p^{n+1} \\ & & \qquad\qquad \text{in } \Omega \\ \nabla\cdot\mathbf{u}^{n+1} & = & 0 \end{array}\right. \tag{2.24}$$

   with the boundary condition $\mathbf{u}^{n+1}\cdot\mathbf{n} = 0$ on $\partial\Omega$. This step can be reformulated such that the computation of the pressure and of the velocity can be separated. Indeed, by taking the divergence of the first equation in (2.24), the following problem is obtained for the pressure:

$$\left\{\begin{array}{lcll} \Delta p^{n+1} & = & \dfrac{1}{\delta t}\nabla\cdot\tilde{\mathbf{u}}^{n+1} & \text{in } \Omega \\ \dfrac{\partial p^n}{\partial \mathbf{n}} & = & 0 & \text{on } \partial\Omega \end{array}\right. \tag{2.25}$$

   The end-of-step velocity can now be determined by:

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}}^{n+1} - \delta t\nabla p^{n+1} \tag{2.26}$$

Note that the conservation of mass is ensured in this method since $\mathbf{u}^{n+1}$ plays the role of the solenoidal part in the Ladyzhenskaya theorem. This method (and all its variations) is perhaps the most widely used method for solving the incompressible fluid flow equations. The main advantage of this type of methods is the reduced computational cost of time-dependent, nonstationary incompressible flows at higher Reynolds number. In other words, their computational efficiency for problems of practical importance. Another advantage of this method in the Finite Element context, is the fact that the LBB-condition (stability condition between the functional spaces containing the velocity and the pressure) needs not to be satisfied. This allows an implementation using equal order elements for velocity and pressure.
However, the projection method has some drawbacks. The main one is perhaps the specification of boundary conditions. First of all, a nonphysical boundary condition is needed in order to solve the pressure Poisson equation (2.25) and the choice of this condition is not as obvious as its seems to be [43]. Second of all, as pointed out by several authors [26], [29], the end-of-step velocity field does not satisfy the intended boundary conditions: the tangential component of the boundary condition cannot be controlled as a consequence of Ladyzhenskaya theorem where only the normal component of the velocity can be prescribed. Another potential drawback of the projection method is the apparition of a spurious numerical pressure boundary layer introduced by the nonphysical boundary condition imposed in (2.25).

Algebraic Splitting
Algebraic splitting methods [37], [64], [71] can be seen as the discrete counterpart of the projection method. They are based on approximate LU-factorisation of the system (2.8) obtained from the discretisation of the flow equations. The system (2.8) can be put into the form:

$$\begin{bmatrix} C & G \\ D & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u^{n+1}} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{b_u^{n+1}} \\ b_p^{n+1} \end{bmatrix} \tag{2.27}$$

where C is the sum of the mass, convection and stiffness matrices, D is the divergence matrix and G the gradient matrix.
The matrix in (2.27) can be factorised as:

$$\begin{bmatrix} C & 0 \\ D & -DC^{-1}G \end{bmatrix} \begin{bmatrix} I & C^{-1}G \\ 0 & I \end{bmatrix} \tag{2.28}$$

Then setting

$$\begin{bmatrix} \mathbf{\tilde{u}^{n+1}} \\ \tilde{p}^{n+1} \end{bmatrix} = \begin{bmatrix} I & C^{-1}G \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathbf{u^{n+1}} \\ p^{n+1} \end{bmatrix}, \tag{2.29}$$

the system can be solved in 3 steps:

1. $C\mathbf{\tilde{u}^{n+1}} = \mathbf{b_u^{n+1}}$

2. $DC^{-1}Gp^{n+1} = D\tilde{\mathbf{u}}^{n+1} - b_p^{n+1}$ (2.30)

3. $\mathbf{u}^{n+1} = \tilde{\mathbf{u}}^{n+1} - C^{-1}Gp^{n+1}$

The main advantage of this method over the continuous projection method is the fact that all boundary conditions are prescribed on the original, unsplit system (2.28). Therefore no auxiliary or nonphysical boundary conditions are needed for neither the velocity nor the pressure.

However, in order for this type of methods to be used in practical compuations, the expensive inversion (at every time step) of the matrix C should be avoided. Therefore, C^{-1} is usually substituted by some proper approximations H_1 and H_2 in steps 2 and 3 of (2.30):

1. $C\tilde{\mathbf{u}}^{\mathbf{n+1}} = \mathbf{b}_{\mathbf{u}}^{\mathbf{n+1}}$

2. $DH_1Gp^{n+1} = D\tilde{\mathbf{u}}^{n+1} - b_p^{n+1}$ (2.31)

3. $\mathbf{u}^{n+1} = \tilde{\mathbf{u}}^{n+1} - H_2Gp^{n+1}$

Several choices for H_1 and H_2 are possible. Two particular cases have been much studied and used:

- $H_1 = H_2$ introduced by Perot [67] which leaves the continuity equation unchanged and therefore gives a solenoidal solution,
- $H_2 = C^{-1}$ which leaves the momentum equation unchanged [70].

In the context of finite element discretisation, one disadvantage of the algebraic splitting over the projection method is the necessity to satisfy the LBB-condition and therefore the necessity of using mixed-element in the implementation.

Some other pressure-correction approaches

SIMPLE

The SIMPLE (Semi-Implicit Method for Pressure Linked Equations) family of methods introduced by Patankar and Spalding in 1972 [66] is also a possible solution technique for solving the incompressible Navier-Stokes equations. It is based on the same basic idea as the projection method, namely a series of guess-and-correct operations: the velocity components are first computed from the momentum equation using a guessed-pressure field. The pressure and velocity are then corrected so that the continuity equation is satisfied. The main distinction between this method and the projection method is the way in which the pressure and velocity corrections are performed [66], [81], [23]. The SIMPLE procedure can be described as follows:

1. Guess the pressure (p_0) at each grid point;

2. Solve the momentum equation to find the velocity field $\mathbf{u}^* = (u^*, v^*, w^*)^T$;

3. solve the following pressure correction equation: $\Delta p^* = \frac{1}{A}(\nabla \cdot \mathbf{u}^*)$ to find p^* at each grid point, where A is a fictitious time increment;

4. Correct the pressure and velocity according to:

$$p = p_0 + p^* \tag{2.32}$$

$$u = u^* - \frac{A}{2\Delta x}\delta_x p \tag{2.33}$$

$$v = v^* - \frac{A}{2\Delta y}\delta_y p \tag{2.34}$$

$$w = w^* - \frac{A}{2\Delta z}\delta_z p \tag{2.35}$$

   where $\delta_x p$ is the variation of the pressure p in x-direction evaluated at the grid points. Simirlarly for $\delta_y p$ and $\delta_z p$.

5. Replace the previous intermediate values of pressure and velocity p_0 and $\mathbf{u}^*$ by the new corrected values p and $\mathbf{u}$ and return to step 2.

Note that in certain cases the convergence rate of the method is found unsatisfactory. This is due to the fact that pressure correction p^* is somehow overestimated in (2.32) [81]. Therefore equation (2.32) is often replaced with $p = p_0 + \alpha_p p^*$, where α_p is an underrelaxation parameter.

PISO

The PISO (Pressure-Implicit with Splitting of Operators) method is another solution technique. It is also based on a preditor-corrector strategy. Here one predictor step and two corrector step are used [22].

A chart of the different methods presented here is shown figure 2.1.

2.3 Spatial discretisation techniques

Independently of which one of the previous solution techniques is used, a discretisation method has still to be chosen in order to transform the continuous problem into a discrete one. Four discretisation techniques have globally gained acceptance in the CFD community. The Finite Difference method (FDM) was historically the first applied. It was quickly followed by the Finite Element (FEM) and Finite Volume (FVM) Methods and then the Spectral Element Method (SEM).

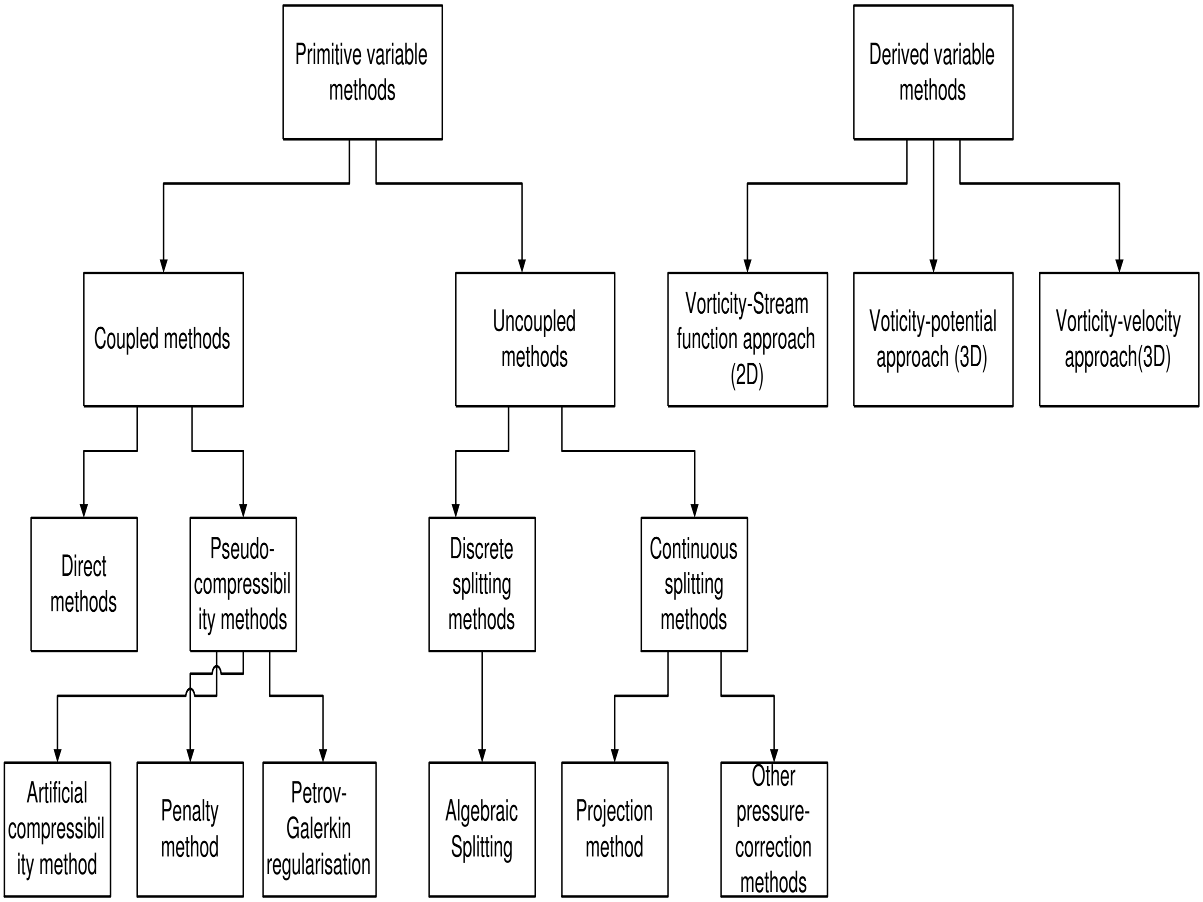

Primitive variable methods
Derived variable methods
Coupled methods
Uncoupled methods
Vorticity-Stream function approach (2D)
Voticity-potential approach (3D)
Vorticity-velocity approach(3D)
Direct methods
Pseudo-compressibil ity methods
Discrete splitting methods
Continuous splitting methods
Artificial compressibil ity method
Penalty method
Petrov-Galerkin regularisation
Algebraic Splitting
Projection method
Other pressure-correction methods

Figure 2.1: Chart flow presenting some of the existing methods for bulk flow.

2.3.1 Finite Difference Method

The philosophy behind the Finite Difference Method is the discretisation of differential operators i.e. to replace the partial derivatives in the equations with algebraic differences within a certain accuracy. This discretisation of the partial differential equations yields to a system of algebraic equations, whose unknowns are the nodal values. We can choose among many different equation solvers (both linear and nonlinear) in order to obtain an approximate solution. There exist many finite difference discretisation schemes, and each of them will give a different algebraic system. Applying, for instance, the second-order centered-space method at the point of coordinates (x_i, y_j) on figure 2.2, the first component of the momentum equation (2.7) and the continuity equation (2.5) for a two-dimensional problem reads to the following semi-discrete equations (for a three-

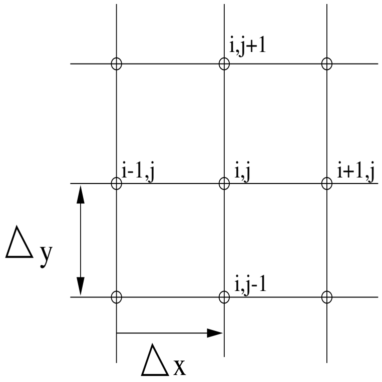


Figure 2.2: Two-dimensional finite difference grid.

dimensional problem, the results are strictly analogous, they only contain more terms):

$$\begin{aligned}\frac{\partial u_{i,j}}{\partial t} + u_{i,j}\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + v_{i,j}\frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y} &= \frac{p_{i+1,j} - p_{i-1,j}}{2\Delta x} \\ \nu\left(\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} - \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2}\right) + f_{1i,j}\end{aligned} \tag{2.36}$$

and

$$0 = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + \frac{v_{i,j+1} - v_{i,j-1}}{2\Delta y} \tag{2.37}$$

where $f_{1i,j}$ is the first component of the body force, u and v are the velocity components in the x- and y-directions respectively, and Δx, Δy are the grid spacings. ν is the kinematic viscosity and p is the kinematic pressure. The subscripts i, j indicates the value of the variables at the grid point (x_i, y_j).

The second component of the momentum equation can be written in a similar manner, and the three-dimensional extension is straight forward.

FDM is usually based on regular meshes and it can be very efficient if the spacing is equidistant and the cells are rectangle (in two-dimensions) or hexaedrons (in three-dimensions). Good results can be obtained when using staggered grids i.e. grids where the pressure is not evaluated at the same nodes as the velocity. The main disadvantages of the FDM are the implementation of boundary conditions for a general geometry and problematic mass conservation properties [51], [81].

2.3.2 Finite Volume Method

In the Finite Volume Method, the conservation principles are applied to a fixed region in space known as control-volume and no longer to nodes as in the FDM. Here, the discretisation is applied to an integral formulation of the equations and is based on balancing

variable fluxes between control-volumes. Therefore, in a finite volume discretisation, the flow characteristics are directly used. The conservation properties are much better than in the FDM case since here both local and global mass conservation can be satisfied in an easy way. Therefore FVM is often seen as the most natural method for treating fluid dynamics problems. Another advantage of FVM over FDM is the treatment of boundary conditions for complex geometry which is less cumbersome. Except for that, the resulting formulae and systems of algebraic equations can be treated as in FDM.
As FVM is based on flux balances between controle volumes, it is common in this methodology to first write the flow governing equations under a more convenient manner:

$$\frac{\partial}{\partial t}\int\int\int_{V(t)} \mathbf{U} dV + \int\int_{S(t)} \mathbf{F_c}\cdot\mathbf{n} dS - \int\int_{S(t)} \mathbf{F_V}\cdot\mathbf{n} dS = \int\int\int_{V(t)} \mathbf{f} dV \tag{2.38}$$

where for an incompressible fluid:

- $\mathbf{U} = \begin{pmatrix} 0 \\ u \\ v \\ w \end{pmatrix}$

- $\mathbf{F_{c1}} = \begin{pmatrix} u \\ u^2+p \\ uv \\ uw \end{pmatrix}, \mathbf{F_{c2}} = \begin{pmatrix} v \\ uv \\ v^2+p \\ vw \end{pmatrix}, \mathbf{F_{c3}} = \begin{pmatrix} w \\ uw \\ vw \\ w^2+p \end{pmatrix}$ are the convective fluxes in the x-, y-, z-directions respectively, and $\mathbf{F_c}\cdot\mathbf{n} = \mathbf{F_{c1}}\cdot\mathbf{n_x}+\mathbf{F_{c2}}\cdot\mathbf{n_y}+\mathbf{F_{c3}}\cdot\mathbf{n_z}$;

- $\mathbf{F_{v1}} = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \end{pmatrix}, \mathbf{F_{v2}} = \nu \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \end{pmatrix}, \mathbf{F_{v3}} = \nu \begin{pmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \end{pmatrix}$ are the viscous fluxes in the x-, y-, z-directions respectively, where $\tau_{ij} = \nu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)$, and $\mathbf{F_v}\cdot\mathbf{n} = \mathbf{F_{v1}}\cdot\mathbf{n_x}+\mathbf{F_{v2}}\cdot\mathbf{n_y}+\mathbf{F_{v3}}\cdot\mathbf{n_z}$;

- $V(t)$ is an arbitrary controle volume enclosed by the surface $S(t)$ and $\mathbf{n}$ is the normal to $S(t)$ pointing outwards.

- $\mathbf{f} = \begin{pmatrix} 0 \\ f_x \\ f_y \\ f_z \end{pmatrix}$ is the body force.

Equation (2.38) is found by integrating (2.5) and (2.7) using the divergence theorem.
The computational domain is divided into controle volumes that can be identified with

an index i. V_i and S_i are then the volume and the total face area of the controle volume i. The first term of equation (2.38) can be expressed as the product of the volume V_i of the cell i and the time derivative of the cell average of the flow variables $\mathbf{U}_i$. This yields to the following semi-discrete form of the governing equations:

$$V_i \frac{\partial \mathbf{U}_i}{\partial t} + \sum_j \mathbf{F}_{\mathbf{c}(i+j)/2} - \sum_j \mathbf{F}_{\mathbf{v}(i+j)/2} = \mathbf{f}_i \tag{2.39}$$

where i is the cell index and j is the index of the next cells of the cell i. *(i+j)/2* denotes the faces between cells i and j as shown in figure 2.3.

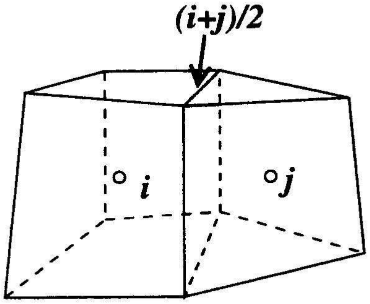


Figure 2.3: Definition of the cell and face index.

Remain to evaluate the convective and viscous fluxes $\mathbf{F_c}$ and $\mathbf{F_v}$. See for example [33], [51], [81], [88], and the references therein for more details about the FVM.

2.3.3 Finite Element Method

In Finite Element Method, the governing equations are written in a variational form before discretisation. This variational formulation corresponds to the minimisation of some energy integral over the domain. The solution of the problem is assumed to have a prescribed form and belongs to a particular functional space. The finite element discretisation consists in the discretisation of that functional space. The integral formulation of the problem confers to the FEM the ability to naturally incorporate differential type of boundary conditions. Moreover, the combination of the representation of the solution in a given function space, with the integral formulation treating rigourously the boundary conditions, gives to the method a rigourous and strong mathematical foundation. This allows a precise definition of accuracy and error analysis, while in FDM and FVM the concept of accuracy is more loosely defined. FEM show its strength when unstructured meshes are needed for complex geometry. However, in FEM mass conservation is satisfied only globally. There is no local mass conservation as it is the case in FVM [69], [28].

One of the most common finite element discretisation is the Galerkin discretisation. If Ω is the computational domain with boundary $\partial\Omega = \partial\Omega_D + \partial\Omega_N$ where $\partial\Omega_D$ and $\partial\Omega_N$ are non-overlapping and with boundary conditions:

$$\mathbf{r}_D(\mathbf{u}) = 0 \quad on \quad \partial\Omega_D \tag{2.40}$$

$$\mathbf{r}_N(\mathbf{u}) = 0 \quad on \quad \partial\Omega_N \tag{2.41}$$

The Galerkin finite element formulation of the problem is: find $u \in V$ and $p \in Q$ such that

$$\int_\Omega \left(\frac{\partial \mathbf{u}}{\partial t}\mathbf{v} + (\mathbf{u}\cdot\nabla)\mathbf{u}\;\mathbf{v} + \nu\nabla\mathbf{u}\cdot\nabla\mathbf{v} - p(\nabla\cdot\mathbf{v}) \right) d\Omega = \int_\Omega f\cdot\mathbf{v}d\Omega + \int_{\partial\Omega_N} \mathbf{r}_N\cdot\mathbf{v}dS \quad \forall\mathbf{v}\in V \tag{2.42}$$

$$\int_\Omega q(\nabla\cdot\mathbf{u})d\Omega = 0 \quad \forall q \in Q \tag{2.43}$$

where

- $V = \{$ all functions $\mathbf{v}$ which are smooth enough and satisfy $\mathbf{v}(\mathbf{x}) = 0 \;\; \forall\mathbf{x}\in\partial\Omega_D \;\}$
- $Q = \{$ all functions q which are smooth enough and satisfy $\int_\Omega q\; d\Omega = 0 \;\}$

The discretisation of the functional spaces V and Q gives the following matrix system to solve:

$$M\frac{\partial \mathbf{u}}{\partial t} + [\nu K\mathbf{u} + L\mathbf{u}] + Gp \;=\; \mathbf{r}_N + \mathbf{b}_u \tag{2.44}$$

$$D\mathbf{u} \;=\; b_p \tag{2.45}$$

where

$$M = [m_{ij}] = \int_\Omega N_i N_j d\Omega \tag{2.46}$$

$$K = [k_{ij}] = \sum_{s=1}^{d}\int_\Omega \frac{\partial N_i}{\partial x_s}\frac{\partial N_j}{\partial x_s}d\Omega \tag{2.47}$$

$$L = [l_{ij}] = \int_\Omega N_i \sum_{s=1}^{d}\sum_{k=1}^{n_s} N_k \frac{\partial N_j}{\partial x_s}u_{sk}d\Omega \tag{2.48}$$

$$G = [g_{ij}] = -\int_\Omega \frac{\partial N_i}{\partial x_s}\phi_j d\Omega \tag{2.49}$$

$$b_{u_{si}} = \int_\Omega N_i f_{si} d\Omega \tag{2.50}$$

with N and ϕ being the form functions for the velocity and pressure respectively, d being the space dimension and n_s the number of velocity nodes in the s-direction. For more details see for example [28].

2.3.4 Spectral Element method

In the spectral element method, the global procedure is similar to FEM, except that the solution has the form of a series of known functions of spatial coordinates with coefficients to be determinated. The approximation functions are not defined locally, as in FEM, but on the whole domain which puts severe restrictions on the domain geometry. Finally, a set of algebraic equations is obtained for the function coefficients. See for example [69] for more details on the FEM and SEM.

2.4 Time discretisation techniques

In the previous, spatial discretisation was briefly introduced. We now consider temporal discretisation. One possibility is to treat the time as another spatial dimension as in [96]. This approach is however not very widespread for three main reasons [53]:

1. For higher-order schemes, this technique produces extremely large matrix system;
2. For lower-order schemes, the resulting algorithms are the same as in finite difference schemes. As finite difference schemes are easier to derive and more studied, they seem to be more convenient in this context;
3. Time, unlike space, has a definite direction. Therefore, schemes that reflect this hyperbolic character will be most appropriate. This is the case of finite differences.

If it is assumed that the spatial discretisation has already been performed, the remaining problem can be expressed as a system of ordinary differential equations to be solved. This can be written on the form:

$$\frac{d\mathbf{u}}{dt} = \mathbf{r}(\mathbf{u}) \tag{2.51}$$

Time stepping schemes are usually divided into two main groups [53]: *explicit* and *implicit* schemes.

2.4.1 Explicit schemes

Almost all explicit schemes may be recast into an m-stage Runge-Kutta scheme of the form:

$$\Delta\mathbf{u}^{n+i} = \alpha_i \Delta t \mathbf{r}(\mathbf{u} + \Delta\mathbf{u}^{n+i-1}), \quad i = 1, m, \quad \Delta\mathbf{u}^0 = 0 \tag{2.52}$$

The coefficients α_i are chosen according to the desired properties, such as damping or temporal order of accuracy. Some popular choices are:

a) m=1 and α_1=1.0: one-stage or forward Euler scheme;

b) m=2 and α_1=0.5 and α_2=1.0: two-stage scheme.

More examples can be found in [53] and [42].

The main properties of explicit schemes are:

- they allow for an arbitrary order of temporal accuracy;
- they are easy to implement;
- the prescription of boundary condition is relatively simple;
- they are easy to parallelise;
- the time-step is limited by stability constraints, such as the Courant-Friedrichs-Levy (CFL) condition.

2.4.2 Implicit schemes

For implicit schemes, the right-hand side is evaluated somewhere between the present time position t^n and the next time position t^{n+1}:

$$\Delta \mathbf{u}^{n+1} = \mathbf{u}^{n+1} - u^n = \Delta t \, \mathbf{r}(\mathbf{u}^{n+\Theta}) \tag{2.53}$$

The right-hand side is usually linearised:

$$\mathbf{r}^{n+\Theta} = \mathbf{r}^n + \left.\frac{\partial \mathbf{r}}{\partial \mathbf{u}}\right|^n \cdot \Theta \Delta \mathbf{u}^{n+1} = \mathbf{r}^n + A^n \cdot \Theta \Delta \mathbf{u}^{n+1} \tag{2.54}$$

where the Jacobian $\left.\frac{\partial \mathbf{r}}{\partial \mathbf{u}}\right|^n$ is denoted A^n . Equation (2.53) can then be written as:

$$(1 - \Delta t \Theta A^n) \cdot \Delta \mathbf{u}^{n+1} = \mathbf{r}^n \tag{2.55}$$

Popular choices for Θ are:

a) $\Theta = 1.0$: Backward Euler (first order accurate);

b) $\Theta = 0.5$: Crank-Nicholson (second order accurate)

The main properties of implicit schemes are:

- the maximum order of accuracy for unconditionnably stable schemes is two;
- "arbitrary" time-steps can be taken, i.e. the time-step Δt is governed only by accuracy consideration, not by stability. However, one should be careful with the word "arbitrary". Implicit CFD codes are usually run with CFL numbers no larger than 100;
- the solution of (2.55) is usually expensive due to the large system of equations appearing on its left-hand side.

2.4.3 Explicit versus implicit

Implicit solutions are often used for steady-state flow problems because they have the potential of achieving rapid convergence. Rapid convergence implies that force imbalances (i.e. transient) are highly damped in succeeding iterations. When using implicit schemes for time-dependent problems, the damping feature which was advantageous for steady solutions becomes a disadvantage. The reason is that this damping can easily overshadow real transient behaviour. Moreover, for time-dependent problems, implicit schemes are more expensive and more difficult to implement than explicit ones. There are however certain classes of time-dependent problems where implicit schemes pay off:

a) In the cases where:

$$\Delta t|_{phys.relevant} \gg \Delta t_{CFL}$$

This is often the case in incompressible flow computations.

b) When the computational grid contains small elements/cells, distorted elements/cells (close to rigid bodies for instance), difficult surface (free surfaces for example), and so forth.

2.5 Transformation of equations and grids

In practical CFD applications, the physical plane seldom allows the direct use of a uniform, rectangular grid. Such grids are particularly interesting when using the finite difference spatial discretisation.
For example, assume we want to compute the flow around a two-dimensional airfoil as shown in figure 2.4.

It is easily noted some problems with this rectangular grid:

a) Some grid points fall inside the airfoil, where they are completely out of the flow. What values of the flow properties should be prescribed there?

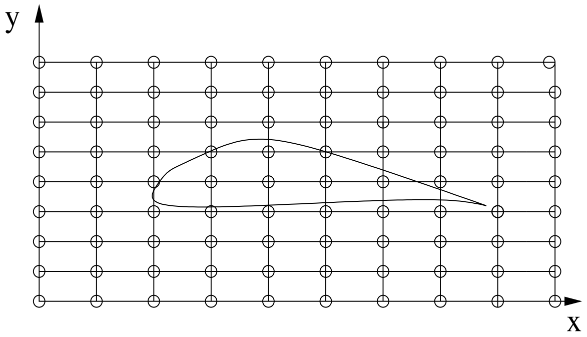


Figure 2.4: Airfoil and rectangular uniform grid in the physical plane.

b) There are a few, if any, grid points that fall on the surface of the airfoil. This is a problem since the airfoil surface is a primordial boundary for the determination of the flow. Therefore the airfoil surface must be strongly modelled for the numerical solution.

The rectangular grid shown in figure 2.4 is therefore not very appropriate for the numerical solution of this problem. In contrast, the non-uniform, curvilinear grid shown in figure 2.5 is more appropriate for this problem. The new coordinates ξ and η are defined such that the airfoil surface corresponds to a coordinate line η=constant. This type of grid is called a boundary fitted grid.

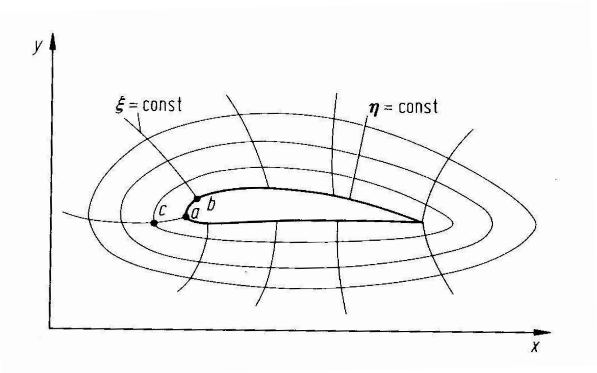


Figure 2.5: Physical plane.

It is worth to enlight two features of this type of grids:

1. The grid points fall naturally on the airfoil surface;

2. Conventional numerical schemes (and in particular finite difference quotients) are difficult to use. The curvilinear physical plane is therefore often transformed into a rectangular computational plane as shown in figure 2.6. This transformation is defined such that there is a one-to-one correspondence the meshes in figure 2.5 and in figure 2.6. For example, points a, b and c in figure 2.5 correspond to points a, b and c in figure 2.6 which involves uniform $\Delta\xi$ and $\Delta\eta$. The computations are performed in the computational plane and the computed information is then transformed back to the physical plane.

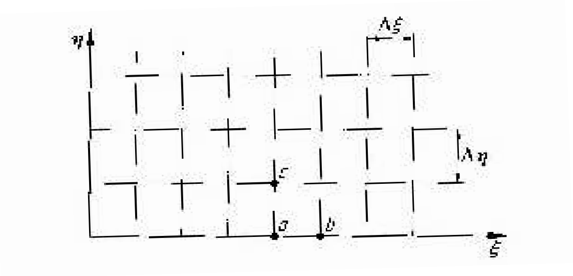


Figure 2.6: Computational plane.

Furthermore, the governing equations which are solved in the computational plane must be expressed in terms of ξ and η and not x and y i.e. the governing equations must be transformed from (x, y, t) to (ξ, η, τ).
For simplicity, we will consider a two-dimensional unsteady flow, with independent variables (x, y, t) in the remaining of this section. The results for three-dimensional unsteady flow, with independent variables (x, y, z, t) are analogous and simply involve more terms. We will only show the principle of the transformation of the equations for more details see for example [92].

The transformation between physical and computational planes is defined by:

$$\xi = \xi(x, y, t) \tag{2.56}$$

$$\eta = \eta(x, y, t) \tag{2.57}$$

$$\tau = \tau(t) \tag{2.58}$$

Using the chain rule of differential calculus, one can write:

$$\frac{\partial}{\partial x} = \left(\frac{\partial}{\partial \xi}\right)\left(\frac{\partial \xi}{\partial x}\right) + \left(\frac{\partial}{\partial \eta}\right)\left(\frac{\partial \eta}{\partial x}\right) + \left(\frac{\partial}{\partial \tau}\right)\underbrace{\left(\frac{\partial \tau}{\partial x}\right)}_{=0} \tag{2.59}$$

$$\frac{\partial}{\partial y} = \left(\frac{\partial}{\partial \xi}\right)\left(\frac{\partial \xi}{\partial y}\right) + \left(\frac{\partial}{\partial \eta}\right)\left(\frac{\partial \eta}{\partial y}\right) + \left(\frac{\partial}{\partial \tau}\right)\underbrace{\left(\frac{\partial \tau}{\partial y}\right)}_{=0} \tag{2.60}$$

$$\frac{\partial}{\partial t} = \left(\frac{\partial}{\partial \xi}\right)\left(\frac{\partial \xi}{\partial t}\right) + \left(\frac{\partial}{\partial \eta}\right)\left(\frac{\partial \eta}{\partial t}\right) + \left(\frac{\partial}{\partial \tau}\right)\left(\frac{d\tau}{dt}\right) \tag{2.61}$$

Equations (2.59), (2.60) and (2.61) allow the derivatives with respect to x, y and t to be expressed in terms of derivatives with respect to ξ, η and τ. Second order derivatives can also be transformed in a similar manner (cf. [92]). Once all the terms in the governing equations have been transformed, numerical schemes can be applied to the equations expressed in ξ, η and τ (cf. [92]).

Chapter 3

Free Surface prediction

The prediction of the free surface is a crucial part in ship hydrodynamics. The position of the surface is known initially and has to be determined as part of the solution at later times. Many methods exist for the prediction of free surfaces. They neither have the same computational efficiency nor can handle the same physical situations (i.e. types of waves). Therefore the choice of the method is very important and must be decided by both the physical problem at hand and the computational resources available.
In this chapter, we will present some of the main techniques used in marine hydrodynamics to predict free surfaces as well as some of their advantages and disadvantages.
These methods can be classified into two main groups:

1. *Interface Tracking methods*: methods which define the free surface as a sharp interface whose motions is followed. The tracking is usually performed by making use of the *kinematic* and *dynamic* free surface boundary conditions. Here boundary fitted grids are used and the grid must be readjusted each time the free surface is moved. The leading question of Interface Tracking methods is: “where is the surface ?”

2. *Interface Capturing* also called *Volume Tracking methods*. Here the computations are performed on a fixed grid, which extends beyond the surface. The shape of the free surface is determined by cells which are partially filled. This is achieved by either following massless particles introduced into the liquid phase near the surface initially or by solving a transport equation either for the void fraction of the liquid phase or for the distance from the interface. Here the leading question is: “where is the fluid volume ? ”

The dominant methods in marine hydrodynamics have been of interface-tracking type. These techniques have worked well for a variety of hull forms (see [46] and [50]) but they have the disadvantage of dissipating the waves out too quickly far from the hull. One way to avoid dissipation (if one needs an accurate description of the free surface away from the hull) is to use RANS combined with potential flow methods or simply potential flow methods away from the hull.

Usually, surface fitted grids are used in interface tracking methods. This allows for a simple and accurate way of prescribing the boundary conditions on the free surface. However surface fitted grids can also be problematic since once the free surface starts changing, the grid must be adjusted in order to accommodate the new free surface height. Here good quality grid is easily lost close to the surface and if the changes are too large the grid can become highly skewed, which is usually a problem for stability of Navier-Stokes solvers. The grid should be readjusted in a manner that automatically generates a "good" grid. Considering how difficult it can be to generate a "good" grid with predefined surfaces, one can easily imagine how difficult it will be to automatically adjust a structured grid to any new position of the water surface for a complex geometry. Unstructured grids have the potential to overcome some of these inherent limitations [52]. However, such complex physical situations like breaking waves cannot be handled.

On the other hand, *volume tracking methods* or *interface capturing methods* can handle such complexities and are receiving an increased attention in ship hydrodynamics. They have been used in combination with both RANS [8], [14], and LES [16], [84, 85].

Historically the first method of this type was the well-known Marker-And-Cell (MAC) method. Then, more than one decade later came the Volume Of Fluid (VOF) method, and finally the level-set method appeared in the late 1980's early 1990's.

Here also the interface evolves as a part of the solution, but no regridding is involved which makes it less cumbersome than techniques where regridding is needed. In *volume tracking methods*, there is a thickness assiocated to the free surface but this can be controlled by the local grid size.

Generally speaking, *interface tracking methods* are more accurate than *volume tracking methods*, and more efficient for very simple physical situations. However, as previously mentioned, *interface tracking methods* are usually limited to simple physical situations. Therefore, it could be ideal that a CFD code used for ship hydrodynamics applications contains both an *interface tracking* and a *volume tracking* method. For simple physical phenomena, the former is preferred because of its higher accuracy while the latter is used when the former cannot be.

We will, in the remaining of this chapter, briefly present some of the most commonly used methods in marine hydrodynamics for predicting free surfaces.

3.1 Height function method

The height function method [61, 62] is a simple but very efficient way, when it can be used, of representing a free boundary. This method is based on treating the free surface directly as a moving boundary and does not deal with volumes of the fluid. It is consequently an *interface tracking method.* Here the free surface position is given by the values of a function h, called *height function*, which is the distance between the free surface and

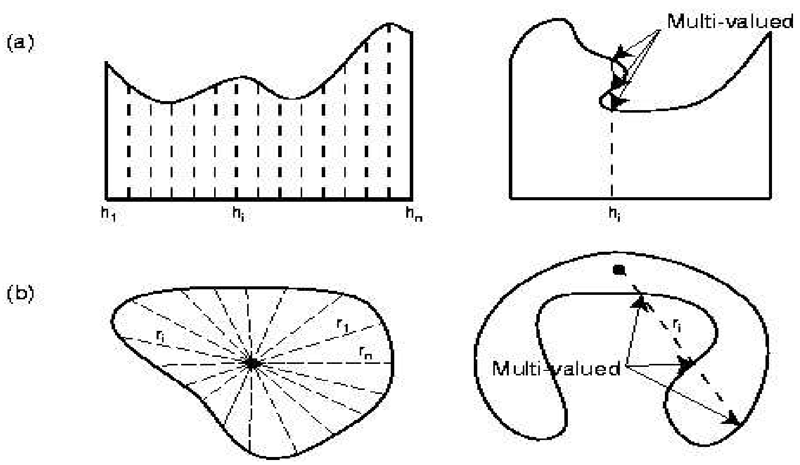


Figure 3.1: Height functions for (a) open interfaces and (b) closed interfaces

a reference line (for two-dimensional problems, cf. fig.3.1) or a reference surface (for a three-dimensional problem). h is then a function of position on the reference surface and can be written as (in a three-dimensional problem):

$$z = h(x, y, t) \tag{3.1}$$

This method works if h is a single-valued function as shown on the left of figure 3.1. This means that physical situations like bubbles, drops, breaking waves cannot be handled by this method. Further, it does not work well when the slope of the free boundary exceeds the mesh cell aspect ratios $\frac{\delta z}{\delta x}$ and $\frac{\delta z}{\delta y}$. However, this method is to be preferred when it can be used because it is simple to implement, extremely robust, computationally efficient and cheap when it comes to storage requirement. Indeed, it requires only a one-dimensional storage array for the free surface height values.
The next step is to choose an equation or a set of equations allowing the calculation of the values of h. The most widely used equations in order to do so are the kinematic and dynamics free surface conditions which in fact are boundary conditions.
In the height function method, the basic procedure for advancing the solution in time consist of three steps:

1. use the current values of the velocity $\mathbf{u^n}$ and of the pressure p^n fields along with the kinematic free surface condition in order to find the new position and shape of the free surface,

2. impose the dynamic free surface condition at the newly computed position of the free surface and solve for the bulk flow,

3. extract the new velocity $\mathbf{u^{n+1}}$ and pressure fields p^{n+1}, and then go back to step 1.

3.1.1 The kinematic free surface condition

In order to derive the kinematic free surface condition let us define the function:

$$F(x, y, z, t) = z - h(x, y, t) \tag{3.2}$$

If we let the free surface being at $z = h(x, y, t)$, then $F = 0$ defines the surface. A fluid particle on the free surface will stay there [47] and will therefore always satisfy the equation $F = 0$. As a consequence, the time rate-of-change of all the particles on the surface will be zero as they are followed in space. This can be written as:

$$\frac{DF}{Dt} = 0 \tag{3.3}$$

where

$$\frac{D()}{Dt} = \frac{\partial()}{\partial t} + \mathbf{u} \cdot \nabla() \tag{3.4}$$

represents the substantial derivative and $\mathbf{u}$ is the velocity of the fluid particle. Equation (3.3) expresses the continuity of the velocity across the surface and can be rewritten as:

$$\frac{\partial}{\partial t}\left[z - h(x, y, t)\right] + \mathbf{u} \cdot \nabla \left[z - h(x, y, t)\right] = 0 \tag{3.5}$$

or again

$$\frac{\partial h(x, y, t)}{\partial t} + u_s \frac{\partial h(x, y, t)}{\partial x} + v_s \frac{\partial h(x, y, t)}{\partial y} - w_s = 0 \tag{3.6}$$

where $\mathbf{u} = (u_s, v_s, w_s)^T$ is the velocity of the fluid at the free surface. Equation (3.6) is the well-known *kinematic free surface condition*. It is used at each time step, to locate the position of the surface.

At that point, the next step is the integration of equation (3.6) in order to obtain the values of h. In order to do so in an efficient way, one needs to know the nature and properties of equation (3.6). Equation (3.6) is hyperbolic. One can then borrow efficient computational techniques from the well-developed field of numerical solutions of hyperbolic conservations laws, in order to integrate equation (3.6) and approximate h. [51] gives more details about numerical solutions of hyperbolic conservation laws.

3.1.2 An integral form of the free surface equation

As mentioned in [51], it might sometimes be preferable, for numerical reasons, to use an integral form of the free surface equation for variable conservation purposes. Several derivation of such a form are possible. One of them is to derive a weak form of equation (3.6) as in [51]. A complete different possibility is to start from the continuity equation (2.5) and integrate it along the vertical (z-axis) from the bottom $z = -B(x, y)$ to the free surface $z = h(x, y, t)$:

$$\int_{-B}^{h} \frac{\partial u}{\partial x} dz + \int_{-B}^{h} \frac{\partial v}{\partial y} dz + \int_{-B}^{h} \frac{\partial w}{\partial z} dz = 0$$

Using Leibniz theorem[1] to switch the order of the differentiation and integration yields to:

$$\begin{aligned} &\frac{\partial}{\partial x} \int_{-B}^{h} u dz + u_b \frac{\partial}{\partial x}(-B) - u_s \frac{\partial h}{\partial x} + \\ &\quad \frac{\partial}{\partial y} \int_{-B}^{h} v dz + v_b \frac{\partial}{\partial y}(-B) - v_s \frac{\partial h}{\partial y} + w_s - w_b = 0 \end{aligned} \tag{3.7}$$

The subscripts b and s indicate values on the bottom and the free surface respectively. The bottom is considered to be a solid boundary and a no-slip boundary condition should be imposed there. The latter boundary condition can be expressed as three equations: one for the normal velocity and two for the tangential velocity. Let us consider the normal component of the the velocity at the bottom. It should satisfy:

$$\mathbf{u}_b \cdot \mathbf{n}_b = 0 \tag{3.8}$$

Equation (3.8) expresses the impermeability of a solid boundary. $\mathbf{u}_b$ is the velocity vector on the bottom, while $\mathbf{n}_b$ is the normal vector to the bottom surface and is a function of x and y. If the bottom is a flat boundary, then the normal $\mathbf{n}_b$ will be constant. If the bottom is not flat, the normal vector must be computed at every point of the bottom surface and possibly (if it is a moving bottom) at every time step. In the case of a non-moving bottom, the normal vector will be computed at the beginning of the simulation once and for all. One way of finding $\mathbf{n}_b$ is to proceed as it was done in order to find equation (3.6). This means defining a function $F(x, y, z)$ that describes the bottom. F can be defined such that its value is zero at all points on the bottom surface. Assuming that the bottom is at $z = -B(x, y)$, it is easy to see that one expression for F is the following one:

$$F(x, y, z) = B(x, y) + z \tag{3.9}$$

[1] Leibniz theorem:

$$\frac{\partial}{\partial x} \int_{a(x,y)}^{b(x,y)} f(x, y, z) dz = \int_{a(x,y)}^{b(x,y)} \frac{\partial f(x, y, z)}{\partial x} dz + f(x, y, b) \frac{\partial b(x, y)}{\partial x} - f(x, y, a) \frac{\partial a(x, y)}{\partial x}$$

The normal vector is then given by:

$$c\left(\frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}, \frac{\partial F}{\partial z}\right)^T = c\left(\frac{\partial B}{\partial x}, \frac{\partial B}{\partial y}, 1\right)^T \tag{3.10}$$

where c is a normalisation constant.
Equation (3.8) can then be written as:

$$u_b\frac{\partial B}{\partial x} + v_b\frac{\partial B}{\partial y} + w_b = 0 \tag{3.11}$$

Inserting equation (3.11) into the integral form of the free surface equation (3.7) leads to:

$$\frac{\partial}{\partial x}\int_{-B(x,y)}^{h(x,y)} udz + \frac{\partial}{\partial y}\int_{-B(x,y)}^{h(x,y)} vdz - \left(u_s\frac{\partial h}{\partial x} + v_s\frac{\partial h}{\partial y} - w_s\right) = 0 \tag{3.12}$$

The free surface must still satisfy the boundary conditions and in particular the kinematic condition (3.6), which after insertion into (3.12) gives:

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}\int_{-B(x,y)}^{h(x,y)} udz + \frac{\partial}{\partial y}\int_{-B(x,y)}^{h(x,y)} vdz = 0 \tag{3.13}$$

One of the main advantages of using equation (3.13) to determine the free surface position is that it accounts for both the kinematic boundary condition and the impermeability at the bottom. This means that equation (3.13) satisfies the conservation of mass criterion. Equation (3.13) is then a conservative form of the free surface equation and might be preferable to use for finite water depth applications.

3.1.3 Moving grid techniques

When using the height function type of methods, it is common to use moving grid techniques in order to solve the kinematic free surface condition (3.6). Descriptions of the method can be found in [32], [36], [19, 20] and [6].
Briefly, the method consists first in casting the flow equations, together with the free surface equations, into a curvilinear coordinate system, more convenient for numerical computations, and then in moving the grid points (on the free surface and in the total- or part of the fluid domain) to fit the new free surface shape [22], [57].

Taking $x_1 = x, x_2 = y, x_3 = z$, using Einstein's convention and following Hinatsu [32], the momentum and continuity equations can be transformed into a free surface fitted curvilinear system (ξ_i, τ) to give

$$\begin{aligned} \frac{\partial}{\partial \tau}\left(\frac{u_i}{J}\right) + \frac{\partial}{\partial \xi_j}\left(\frac{u_i}{J}\frac{\partial \xi_j}{\partial t} + \frac{u_i}{J}\tilde{u}_j\right) & \\ = -\frac{\partial}{\partial \xi_j}\left(\frac{p}{J}\frac{\partial \xi_j}{\partial x_i}\right) + \nu \frac{\partial}{\partial \xi_j}\left(\frac{1}{J} g_{jl}\frac{\partial u_i}{\partial \xi_l}\right) & \qquad (3.14) \\ \frac{\partial}{\partial \xi_j}\left(\frac{\tilde{u}_j}{J}\right) = 0 & \qquad (3.15) \end{aligned}$$

where $J = det\left(\frac{\partial \xi_i}{\partial x_j}\right)$ is the Jacobian of the transformation from the coordinate system x_i to the coordinate system ξ_i, $\tilde{u}_j = \frac{\partial \xi_j}{\partial x_k} u_k$ is the contravariant velocity component along the ξ_j-axis and finally $g_{jl} = \frac{\partial \xi_j}{\partial x_k}\frac{\partial \xi_l}{\partial x_k}$ is the metric tensor.

The kinematic free surface condition (3.6) (as well as the dynamic free surface condition see section 4.2) must also be cast into the free surface fitted coordinate system as the momentum and continuity equations to give

$$\frac{\partial h}{\partial t} + \sum_{j=1}^{j=2} \tilde{u}_j \frac{\partial h}{\partial x_j} = u_3 \qquad (3.16)$$

Farmer et al. [19, 20] used this method in order to compute the fluid flow around a Wigley parabolic hull and a Series 60, $C_b = 0.6$ hull. The results presented for the wave elevation are in good agreement with experimental data. One particularity of Farmer et al.'s algorithm is the existence of temporary leakage of flow through the free surface.

Beddhu et al. [6, 7] used method of technique to also predict flow around a Wigley hull, a Series 60 $C_b = 0.6$ and a Model 5415. Beddhu et al.'s algorithm differs from Farmer et al.'s one in three aspects. First of all there is no leakage of the flow though the surface. Secondly, Beddhu et al. used a background grid whose only purpose is to simplify the grid regeneration process once the free surface evolves. This background grid is generated by extending the free surface blocks of the computational grid beyond the free surface (blocks with the free surface as a boundary are called free surface blocks). Finally, in Beddhu et al.'s algorithm the curvilinear coordinate system is introduced on a curved surface as opposed to a flat surface as in Farmer et al.'s. This gives to the algorithm the ability to model breaking waves to the point of re-entry.

Farmer et al. [19] note that this type of method usually suits to inviscid computations better than to viscous ones when computing flow around surface piercing bodies. This is due to the fact that according to the kinematic free surface condition, the flow should be tangential to the free surface. In inviscid computations, tangent flow along a rigid wall can easily be enforced. However, this is not the case for viscous computations since the no-slip

boundary condition at rigid walls is inconsistent with the free surface boundary condition at the wall/waterline intersection. One way to circumvent this difficulty is to evaluate the water elevation on the wall by extrapolating the value inside the fluid domain.
Löhner et al. [52] also use moving grid techniques to compute the flow field around various hull forms. However, in contrary to the previous authors they do not update the grid at every time step but only every 100-250 time steps. This leads to a minimisation of the computational cost associated to geometry recalculation and grid repositioning.

3.2 Line segment method

The line segment method [60] is a generalisation of the height function method which uses chains of short line segments in order to represent the free surface. The length of these lines must be less than the minimum mesh size (δx, δy, or δz) for accuracy purposes. Each line is represented by its two endpoints, and the coordinates of each point must be stored. Therefore, the storage requirement is slightly increased compared with the height function method, but the method is not limited to single-valued surfaces.
The evolution of the free surface in time is accomplished by moving each line segment, or more precisely the endpoints of each segment, with the local fluid velocity, determined by interpolation of the surrounding mesh.
There are however two main difficulties with this method. The first one is when surfaces intersect or when a surface folds over itself. A special scheme is needed to first detect such situations and secondly to reorder the segments. This is in general not a trivial task. The second main difficulty is the extension of the methods to three-dimensional surfaces. In three dimensions, the determination of neighbouring points defining the local surface configuration requires a large effort. Moreover, the determination of surface intersections and the reordering process is considerably more complex than in two dimensions. Therefore this technique is almost exclusively applied to two-dimensional problems, and very few three-dimensional applications can be found in the literature.

Both the height function and the line segment methods are *interface tracking* type of methods. But the height function method is usually preferred because of its simplicity and of its higher computational efficiency.

3.3 Marker-And-Cell method

The *Marker-And-Cell* technique of Harlow and Welch is a *volume tracking* type of method which uses particles with no mass nor energy, distributed in the whole fluid volume to trace the free surface [30] (see figure 3.2). Those massless particles are fictitious, of Lagrangian type, play no role in the dynamics of the fluid and are not accounted for in the solution

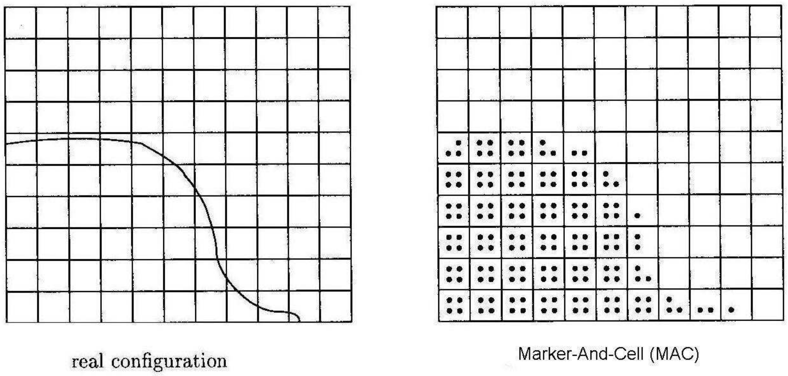


Figure 3.2: Representation of an interface by the Marker And Cell method.

of the flow governing equations. Initially distributed in the whole fluid volume, they are moved passively with the local fluid velocity. The instantaneous surface configuration, can be determined by finding the markers' position after advection.

The computational procedure is very simple and consists of two main steps:

1. the governing equations of the flow are solved on a fixed computational grid and the velocity field is determined inside the fluid domain.

2. each particle is moved according to the velocity at its position.

The fixed computational mesh covers the whole area of possible fluid movement, so that some cells might remain empty. Cells with no marker particles are considered to be empty. Cells with marker particles and with at least one empty neighbour are on the free surface and are called *boundary cells*. Finally, cells with marker particles and no empty neighbour are considered to be filled with fluid.

The method has been much studied and utilised and consequently has been improved in various ways. For instance, by Miyata et al. [54], Miyata [56], Tomé and McKee [86] in the framework of the finite difference method or again by Nakayama and Mori [59] in the context of finite elements. All these authors showed computations in good agreement with experimental data for two-dimensional problems.

One great advantage of this method is its ability to handle complex and general situations as breaking surfaces, dam breaking, splash, or fluid detachment. This method is however computationally prohibitive for large scale three-dimensional applications because of the need to use a large number of continually redistributed particles in order to capture the free surface shape properly. The markers need to be continually redistributed because

their distribution varies with time due to velocity gradients, inflows, outflows, and it is therefore necessary to redistribute the markers evenly in all fluid cells when they tend to spread disproportionally. Further, the storage requirement is significantly high since a very large number of point coordinates must be stored in addition to the bulk flow grid points.
Moreover, even with a large number of markers, it is difficult to determine the orientation of the surface in a cell. The fact that the number of particles is finite can lead to a new problem: false regions of void can be generated in regions with large velocity gradients.
Another problem is the difficulty to impose boundary condition on the surface and especially for the pressure as first noted by Harlow & Welch [30].

3.4 Volume of fluid method

The *Volume of Fluid* (VOF) method is another volume tracking type of technique. It was originally introduced in 1981 by Hirt & Nichols [35] and can be applied to problems where several fluids with different densities are present. It can handle complex physical situations as breaking surfaces, splash, fluid detachment, etc... (see figure 3.3).

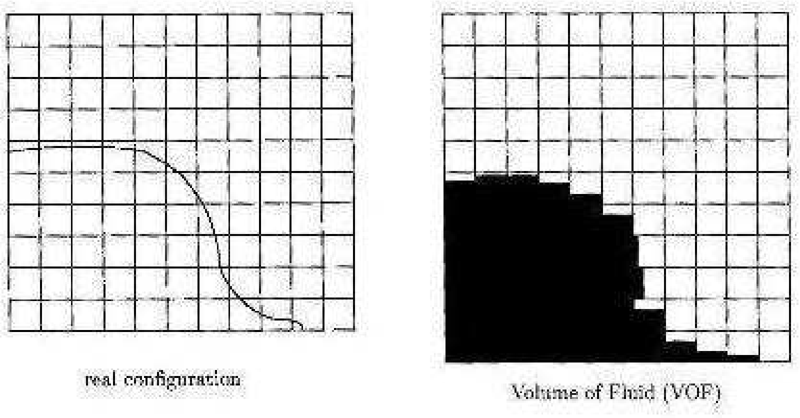
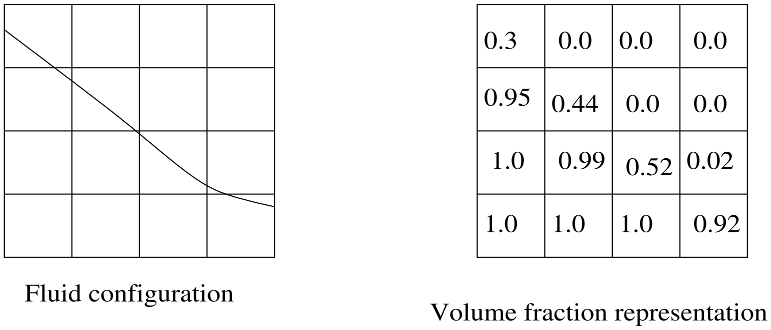


Figure 3.3: Example of interface reconstruction with the VOF method.

The main idea of the VOF method is to introduce a function ϕ whose value is one at any point occupied by fluid and zero otherwise. The average value of ϕ in a cell will then represent the fractional volume of the cell occupied by the fluid. In particular $\phi = 1$ corresponds to cell filled with fluid, $\phi = 0$ to a cell with no fluid at all, and $0 < \phi < 1$ to cell containing the free surface (see figure 3.4).



Fluid configuration

0.3	0.0	0.0	0.0
0.95	0.44	0.0	0.0
1.0	0.99	0.52	0.02
1.0	1.0	1.0	0.92

Volume fraction representation

Figure 3.4: Volume fraction on a discrete mesh.

The time dependence of ϕ is governed by the following pure advection equation [35]:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = 0 \tag{3.17}$$

A typical VOF algorithm generally consists of two parts:

1. a device to track the volume and locate the free surface. This device must be able to keep the interface as sharp as possible,

2. a way to impose boundary conditions at the surface.

The first part of a VOF algorithm, i.e. the location of the free surface consists in a device allowing the accurate computation of the evolution of the function ϕ. This means solving equation (3.17). This is usually done on a grid which extends beyond the fluid domain. As a consequence, air flow can also be computed.
Several approaches have been used. In [35], Hirt & Nichols used a finite difference scheme in order to compute the values of ϕ using a modified equation obtained by combining equation (3.17) with the continuity equation (2.5):

$$\frac{\partial \phi}{\partial t} + \frac{\partial (\phi u)}{\partial x} + \frac{\partial (\phi v)}{\partial y} + \frac{\partial (\phi w)}{\partial z} = 0 \tag{3.18}$$

where u, v, w are the components of the velocity filed $\mathbf{v}$ in x, y, z-direction respectively. Azcueta et al. [1, 2] and Muzaferija & Perić [58] solved the following integral form of equation (3.17) using finite volume discretisation:

$$\frac{d}{dt}\int_V \phi dV + \int_S \phi \mathbf{u} \cdot \mathbf{n} dS = 0 \tag{3.19}$$

In any case, flow rates at cells' interfaces must be computed. These flow rates usually depends on several geometrical factors. For instance, the flow rate Q^n_{ik} through the face k of cell i at time t_n can be written as:

$$Q_{ik}^n = -\alpha_k \int_k u_j^n n_j dS \tag{3.20}$$

where α_k is the wet fraction of face k, u_j^n is the velocity field at face k and n_j is the unit normal vector.
An important aspect of the first part in the algorithm is a scheme preventing an important smearing of the surface. The critical issue here is the discretisation of the convective term in equation (3.17). The problem comes from the fact that the free surface is represented by a discontinuity in the values of the volume fraction ϕ. Indeed, if classical high-order schemes are used, unphysical oscillations will appear in the vicinity of the surface. But on the other hand, if low-order schemes are used, then numerical diffusion will be introduced, leading to a smearing of the free surface over several cells, typically one to three cells. Grid refinement and numerical schemes with high resolution are therefore important parameters for an accurate resolution of the free surface. Chen et al. [11] developed a grid refinement technique called the *surface marker and micro cell method* using a simple refinement criterion: each cell with a value of $\phi \in]0, 1[$ need to be refined. High resolution schemes can be "borrowed" from the area of numerical solutions of hyperbolic conservations laws, and ϕ can be approximated accurately (see [51] and the references therein for more details about numerical solutions of hyperbolic equations) since equation (3.17) is an hyperbolic equation and the function.

It is also possible to use other approaches than those described in [51] in order to obtain Q_{ik}^n. A number of techniques based on purely geometrical considerations have been developed and used with the Volume Of Fluid method. Rudman [74] reviews and discusses advantages and disadvantages of some the existing schemes: the donor-acceptor algorithm as in [72], [35] and [21], the SLIC algorithm [63], the FCT-VOF [95, 74] and Youngs' method [94, 74]. All these methods have been applied to hydrodynamics problems and have given reliable results [21], [35]. Others methods, not included in Rudman's review [74], also give good results for the computation of free surface flows. The high-resolution interface-capturing scheme (HRIC) of Muzaferija & Perić [58] was used by Azcueta et al. [1, 2] to compute both the flow around hydrofoils under the free surface and the breaking bow waves for a ship hull. Kawamura & Miyata [44] developed the *density function method* (product of the density and VOF function ϕ) and located the free surface as the iso-contour of value 0.5. Both the liquid and gas flow are computed here and, the free surface is treated as a boundary where the kinematic and dynamic boundary conditions have to be prescribed.

The second part of a VOF algorithm is a way to impose boundary conditions on the free surface. The grid points will usually not correspond to points on the free surface. Figure 3.5 shows an example of this situation for a two-dimensional application. Here, the shaded part represents the fluid domain. The computational grid is shown only for

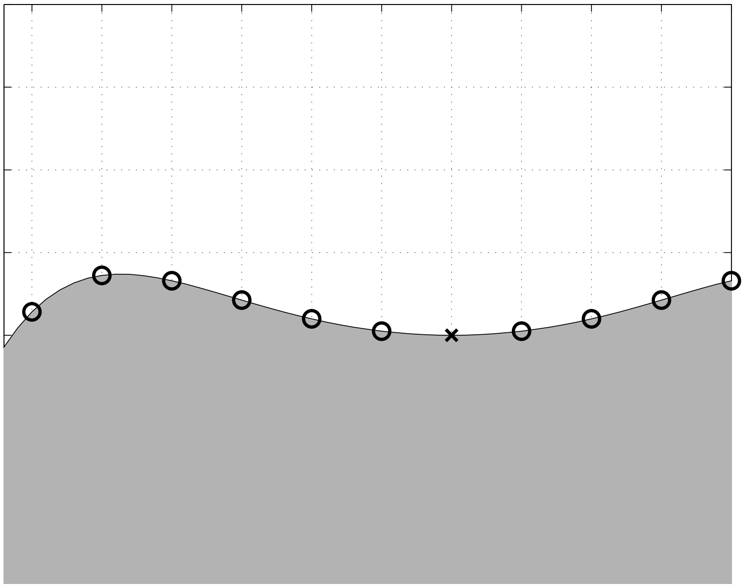


Figure 3.5: Prescription of the boundary conditions in the VOF method.

the air domain. The circles indicate points on the free surface that are not grid points while the cross indicates a point on the free surface that is a grid point. From figure 3.5, one can easily understand that the prescription of boundary conditions on the exact free surface location can become a problem. In [35], where the momentum and continuity equations are solved for the liquid phase only, Hirt and Nichols used a pressure interpolation method, developed originally for the MAC method [60], in order to prescribe the free surface conditions at the correct free surface location. Alternatively, Muzaferija and Perić [58] treated both fluids as a single one whose properties vary in space according to the volume fraction of each of each phase:

$$\rho = \rho_1 c + \rho_2 (1-c), \qquad \mu = \mu_1 c + \mu_2 (1-c) \tag{3.21}$$

where subscripts 1 and 2 denote the two fluids. In this method, the interface is not treated as a boundary and therefore no boundary condition need to be prescribed on it. The interface is simply the location where the fluid properties change abruptly.

The VOF method requires only a one-dimensional array (the value of ϕ in each cell) storage as in the height function method. Furthermore, because it follows regions and not surfaces, all problems associated with intersecting surfaces or surfaces folding over themselves are avoided with the VOF technique. However, the computational cost is increased with respect to the height function technique (but decreased with respect to the MAC method).

Volume of Fluid methods have been much used for marine hydrodynamics applications. Primarily for internal flows as shloshing in tanks, but it has also lately been applied to external flows problems like flows around ship hulls. Azcueta et al. [2] studied an immersed hydrofoil close to the free surface with both an height function method and volume of fluid method. They showed good results for breaking waves situations with the volume of fluid method, while the height function technique was limited to non-breaking wave situations. Azcueta et al. [1], showed that the VOF method can be applied successfully to complex wave phenomena around ship hull, surface-piercing or submerged bodies. Fekken et al. [21] showed that the VOF method can describe quite well the global behaviour of the water surface for green water situations.

3.5 Level Set method

The level set technique was first introduced by Osher and Sethian [65]. A general description of the method can be found in [77]. It is, as in the VOF method, based on the solution of following transport equation:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = 0 \tag{3.22}$$

However, the meaning of the unknown function ϕ is here slightly different from the VOF method. While in the latter ϕ represented the fractional volume of liquid contained in a cell, here it represents the signed distance between a point and the interface. The value of ϕ at a point $\mathbf{x}$ indicates if $\mathbf{x}$ is in liquid or gas as follows:

$$\begin{gathered} \phi > 0 \rightarrow \text{Gas} \\ \phi = 0 \rightarrow \text{Surface} \\ \phi < 0 \rightarrow \text{Liquid} \end{gathered}$$

The level set technique is often classified as an interface-capturing scheme. This might be due to the fact that in the earliest level set algorithms, the zero level set of ϕ was never found explicitly [78]. However, it is probably better to classify the more recent level set methods [80] as “Eulerian” interface-tracking methods. There, interfaces are no longer captured but tracked since the zero level set of ϕ is found explicitly as a part of the numerical algorithm. The term “Eulerian” refers then to the fact that the interface is not represented as a set of connected moving points but as an embedded boundary (the zero level set of the function ϕ).

The level set technique can, as the VOF method, handle complex physical situations like breaking waves, droplets, fluid detachment, etc... Surface tension can be included in a simple way. Coupled with the momentum and continuity equations, equation (3.22) can be solved to give the interface shape and location.

Typically, a level set algorithm consists firstly of a device solving equation (3.22) and secondly of a re-initialisation scheme.

Equation (3.22) can be solved in various ways. Sussman et al. [78, 79], Bet et al. [9], Cura Hochbaum & Schumann [15] and Vogt & Larsson [91] use a two-phase flow formulation. They compute both the water and air flow. The free surface is not sharply resolved but it is given a finite thickness in terms of number of cells where the fluid properties change smoothly. Therefore no free surface boundary conditions need to be prescribed. The introduction of the transition region between the two fluids can be simply seen as a numerical device avoiding the numerical difficulties represented by the abrupt change in density and viscosity between the two fluids, and by the Dirac delta function contained in the expression of the surface tension when it is accounted for.

Following Chang et al. [10], the surface tension can be written as:

$$-\tau\kappa(\phi)\delta(\phi)\nabla\phi \tag{3.23}$$

where

- τ is the coefficient of surface tension,
- $\kappa(\phi) = \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|}\right)$ is the curvature of the interface,
- $\delta(\phi) = 1$ if $\phi = 0$, and 0 otherwise, is the Dirac delta function.

The surface tension is important, and should be accounted for, when the Weber number ($We = \frac{\rho U^2 L}{\tau}$, where U is a characteristic velocity and L a characteristic length) is of order 1 or smaller. This usually happens when the size of curvature of the interface is of the same order of magnitude as the liquid depth. Typical examples are droplets, ripple waves. If We is very large, its effect can be neglected.

The finite thickness mentioned previously, is introduced through a smoothing of the Dirac delta function $\delta(\phi)$ and the introduction of a smooth Heavyside function H_ϵ in the formulations of the density ρ and viscosity μ functions, so that the change between the gas (ρ_g, μ_g) and the liquid properties (ρ_l, μ_l) happens in a band of finite thickness. Many expressions for the regularised Heavyside function have been used. One of them [79] is:

$$H_\epsilon(\phi) = \begin{cases} 0 & \text{if } \phi < -\epsilon \\ \frac{1}{2}[1 + \frac{\phi}{\epsilon} - \frac{1}{\pi}sin(\frac{\pi\phi}{\epsilon})] & \text{if } |\phi| < \epsilon \\ 1 & \text{if } \phi > -\epsilon \end{cases} \tag{3.24}$$

The Dirac delta function in the expression of the surface tension (3.23) is replaced by:

$$\delta_\epsilon(\phi) = \frac{dH_\epsilon}{d\phi} \tag{3.25}$$

The thickness of the interface can be regulated through the parameter ϵ.
The smooth density and viscosity are then written as:

$$\rho_\epsilon(\phi) = \rho_g + (\rho_l - \rho_g)H_\epsilon, \;\; \mu_\epsilon(\phi) = \mu_g + (\mu_l - \mu_g)H_\epsilon \tag{3.26}$$

The regularised surface tension can then be written as:

$$-\tau\kappa(\phi)\delta_\epsilon(\phi)\nabla\phi \tag{3.27}$$

However, in ship hydrodynamics the calculations mostly concern the water flow and the fact that introducing a band through which the physical properties change smoothly is an unphysical approximation. It is therefore favorable to keep the computations in the air domain as well as the band width to a minimum. This led Vogt & Larsson [91] to develop an one-phase formulation of the level set method where only the water flow is computed and where the free surface is represented as a sharp interface. Boundary conditions on the free surface are then needed and Vogt & Larsson use an interpolation/extrapolation scheme in order to prescribe the dynamic boundary condition on the free surface. Vogt & Larsson [91] concluded however that this formulation is so far less flexible and more complicated to implement (particularly in three-dimensions) than the two-phase formulation. They also noted that the one-phase formulation is generally slightly less accurate.

Sussman & Dommermuth proposed a coupled level set and volume of fluid method and applied it to the flow around a DDG5415. They solve the transport equation (3.17,3.22) for both the level set function (denoted by ϕ) and the volume fraction of liquid in each cell (denoted by F). After ϕ^{n+1} and F^{n+1} have been updated, the VOF reconstructed surface is used in the re-initialisation step of the level set function. Their re-initialisation step replaces the current value of ϕ^{n+1} with the exact distance to the VOF reconstructed interface.

The re-initialisation step of a level set algorithm is needed in order to insure that the level set function ϕ remains a distance function through the computations. This is crucial to keep the interface thickness constant in time. ϕ generally does not remain a distance function under the evolution of equation (3.22). The re-initialisation step is then simply the replacement of ϕ^n with $\tilde{\phi}^n$ which in each point represents the distance to the interface and will be use as starting point for the next time iteration in the solution of (3.22). The function $\tilde{\phi}^n$ is usually found as the steady state solution of the following equation:

$$\frac{\partial\tilde{\phi}}{\partial\tilde{t}} + S(\phi)\left[1 - |\nabla\tilde{\phi}|\right] = 0 \tag{3.28}$$

where $\tilde{t}$ is an artificial time and $S(\phi)$ is

- either the sign function as in [79], [41]:

$$S(\phi) = \begin{cases} -1 & \text{if } \phi < 0 \\ 0 & \text{if } \phi = 0 \\ 1 & \text{if } \phi > 0 \end{cases} \tag{3.29}$$

- or a smooth sign function as in [15], [91]:

$$S(\phi) = \frac{\phi}{\sqrt{\phi^2 + \epsilon^2}} \tag{3.30}$$

Equation (3.28) is a nonlinear Hamilton-Jacobi type of equation, with discontinuous coefficients if (3.29) used or continuous coefficients if (3.30) is used. Efficient and accurate solution techniques can then be borrowed from the theory of hyperbolic conservation laws.

The level-set technique has gained a large interest in the ship hydrodynamics community the last decade. As an example of application, Bet et al. [9] applied it with and artificial compressibility technique and finite volume discretisation to compute the flow field around a Wigley hull and a Series 60, both in deep and restricted waters. In [14], Chun et al. also computed the flow around a Wigley hull and a Series 60 in shallow waters. In [91], the flow around a submerged NACA0012 hydrofoil is computed.

3.6 Volume of Fluid versus Level-set

- Note that if in equation (3.5), we set $z - h(x, y, t) = \phi$, then equation (3.5) can be written as:

  $$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0 \tag{3.31}$$

  For an incompressible fluid this is equivalent to

  $$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = 0 \tag{3.32}$$

  since $\nabla \cdot \mathbf{u} = 0$. The previous equation is the transport equation solved in both the level set and the volume of fluid methods. It is then clear that the volume of fluid and level set methods satisfy implicitly the kinematic free surface condition.

- Both techniques are of Eulerian nature, which make them very attractive compared to Lagrangian techniques since they avoid many of the Lagrangian and topological change problems.

- Due to the continuity of the level-set function ϕ and to the introduction of the regularised Heavyside function, no particular difficulties arise in solving the transport equation (3.22). This not the case in the VOF method where some special treatment of the discontinuity is needed. As a consequence of this, level-set is usually more accurate than VOF: higher order (and in particular second order for typical practical computations) schemes can more easily be applied in the level-set technique. Indeed, as mentioned in section 3.4 a second order scheme for hyperbolic conservation laws gives (if no particular treatment is done) oscillations near discontinuities due to its dispersive behaviour. Since in the VOF method the free surface is represented as a discontinuity (of the volume fraction), oscillations close to the surface will appear. If first order schemes are used then there will be an important smearing of the interface, due to the dissipative behaviour of first order schemes.

- Considerable work may be required to develop higher order versions of VOF schemes.

- Calculation of intrinsic geometric properties of the surface, such as curvature or normal direction may be inaccurate in VOF. Even though the definition of those characteristic geometric properties is exactly the same in both cases, $\mathbf{n} = \dfrac{\nabla\phi}{|\nabla\phi|}$ for the normal to the surface and $\kappa = \nabla \cdot \dfrac{\nabla\phi}{|\nabla\phi|}$ for the curvature, these are easier to compute in the case of the level-set technique where the function ϕ is continuous across the surface.

- Thomas et al. [84, 85] pointed out that VOF method automatically satisfies the conservation of mass and momentum but not the conservation of energy while, according to Rudman [74] the level set technique does not guarantee mass conservation in highly distorted flows and this can give unacceptable errors in the results.

- The re-initialisation process is also an important source of errors in the level set methodology.

3.7 Arbitrary Lagrangian-Eulerian methods

As mentioned above, purely Lagrangian methods are limited by their ability to easily handle the strong distortions which often characterise flows of interest. On the other hand, Eulerian methods can relatively easily cope with those distortions but at the expense of a accurate definition of the interface. The Arbitrary Lagrangian Eulerian (ALE) formulation can be considered as a generalisation of of both Lagrangian and Eulerian formulations, attempting to overcome their respective weaknesses.

In the ALE description, the computational mesh is treated as a reference frame moving with an arbitrary velocity $\mathbf{w}$. This velocity $\mathbf{w}$ basic link with the fluid velocity $\mathbf{u}$. Three cases can be enlightened, depending on the value of $\mathbf{w}$:

(a) $\mathbf{w} = 0$, the reference frame is fixed and this corresponds to the eulerian description;

(b) $\mathbf{w} = \mathbf{u}$, the reference frame moves in space with the same velocity as the fluid: this corresponds to the Lagrangian description;

(c) $\mathbf{w} \neq 0$ and $\mathbf{w} \neq \mathbf{u}$, the reference frame moves in space with a velocity which is different from the fluid velocity. This is the ALE description.

Donea et al. [17] present a general description of the ALE formulation of the governing equations.

A summary of the different methods presented here is shown figure 3.6.

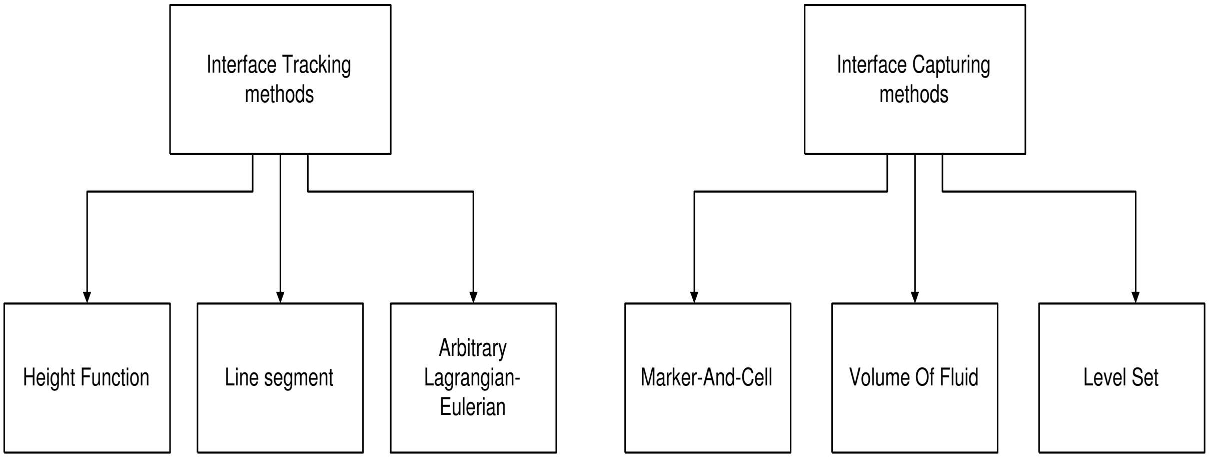

Interface Tracking methods
Interface Capturing methods
Height Function
Line segment
Arbitrary Lagrangian-Eulerian
Marker-And-Cell
Volume Of Fluid
Level Set

Figure 3.6: Chart flow presenting some of the existing methods for prediction of interfaces.

Chapter 4

Initial and Boundary conditions

In this section, initial and boundary conditions will be described briefly. Proper initial and boundary conditions must be provided in order to have a well-posed problem, so that the partial differential equations governing the flow have a unique solution which depends continuously on the data.
The computational domain $\Omega \in I\!R^3$ will generally be bounded by a boundary Γ composed by the bottom, by moving free surface, by lateral impenetrable (side boundaries, body) and open (inflow, outflow) boundaries.

4.1 Initial conditions

The initial conditions should be provided for all the points inside the computational domain as well as those on the boundary for all independent variables such as velocity and free surface elevation. For incompressible flow it is highly important that the initial velocity field $\mathbf{u}_o$ is solenoidal and fulfills the boundary conditions [26] :

$$\mathbf{u}(\mathbf{x},t) = \mathbf{u}_o(\mathbf{x},0) \quad \text{for } \mathbf{x} \in \Omega \cup \Gamma \quad \text{and } t = 0 \tag{4.1}$$

$$\nabla \cdot \mathbf{u}_0 = 0 \quad \text{for } \mathbf{x} \in \Omega \quad \text{and } t = 0 \tag{4.2}$$

$$\mathbf{u}_0 \cdot \mathbf{n} = \mathbf{w}(\mathbf{x},0) \cdot \mathbf{n} \quad \text{for } \mathbf{x} \in \Gamma \quad \text{and } t = 0 \tag{4.3}$$

where $\mathbf{w}$ is a known function.
It is of common practise to set the initial velocity field inside the domain to zero. However other choices are possible like for instance the value of the inflow velocity.

4.2 Dynamic condition

The dynamic condition is a boundary condition that can be prescribed on all surfaces where stress continuity is satisfied. This is for instance the case at the free surface. Once

the shape of the free surface has been found, the dynamic condition is imposed as a boundary condition for solving boundary condition in the Navier-Stokes equations for the bulk flow.
This dynamic free surface condition is obtained from the continuity of the stress vector :

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \boldsymbol{\sigma}_{in} \cdot \mathbf{n} = \boldsymbol{\sigma}_{out} \cdot \mathbf{n} \tag{4.4}$$

where $\mathbf{n}$ is the surface normal and $\boldsymbol{\sigma}$ the stress tensor. Owing that the free surface can be represented by the equation $F(x, y, z, t) = 0$ (cf. equation (3.2) and section 3.1.1), the surface normal is simply given by :

$$\mathbf{n} = \frac{\nabla F}{|\nabla F|}$$

The normal vector points out of the fluid domain.
The inner and outer stress tensor are respectively given by :

$$\boldsymbol{\sigma}_{in} = (-p_{in}\mathbf{I} + 2\mu\mathbf{S}) \tag{4.5}$$

$$\boldsymbol{\sigma}_{out} = (-p_{out}\mathbf{I} + \boldsymbol{\tau}) \tag{4.6}$$

where p_{in} and p_{out} are respectively the pressure in the fluid and the air, $\mathbf{I}$ is the identity matrix, $\mathbf{S}$ the rate-of-strain tensor, μ the dynamic viscosity $\boldsymbol{\tau}$ the outside boundary stress tensor which contains the surface tension. Surface tension effects are important and must be accounted for if the Weber number We $(= \dfrac{\rho_1 L U^2}{\sigma})$ is of order 1, or smaller than 1.
The rate-of-strain tensor $\mathbf{S}$ is defined as :

$$S_{ij} = \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$$

The stress vectors $\boldsymbol{\sigma}_{in}$ and $\boldsymbol{\sigma}_{out}$ can be projected onto the normal direction to the surface, and onto the tangential directions. This gives three conditions:

$$\begin{aligned} \sigma_n &= (\boldsymbol{\sigma} \cdot \mathbf{n}) \cdot \mathbf{n} \\ \sigma_t &= (\boldsymbol{\sigma} \cdot \mathbf{n}) \cdot \mathbf{t} \end{aligned} \tag{4.7}$$

After some algebraic manipulations, one can find :

$$\begin{aligned} \sigma_n &= -p_{in} + 2\mu \frac{\partial u_n}{\partial \mathbf{n}} \\ \sigma_t &= 2\mu \left(\frac{\partial u_n}{\partial \mathbf{t}} + \frac{\partial u_t}{\partial \mathbf{n}} \right) \end{aligned} \tag{4.8}$$

The tangential projection σ_t consists of two components.
The dynamic free surface condition can then be written as :

$$
\begin{aligned}
-p_{in} + 2\mu \frac{\partial u_n}{\partial \mathbf{n}} &= -p_{out} + \tau_n \\
2\mu \left(\frac{\partial u_n}{\partial \mathbf{t}} + \frac{\partial u_t}{\partial \mathbf{n}} \right) &= \tau_t
\end{aligned}
\tag{4.9}
$$

By neglecting the surface tension a simplified set of dynaimc boundary conditions is obtained :

$$
\begin{aligned}
-p_{in} + 2\mu \frac{\partial u_n}{\partial \mathbf{n}} &= -p_{out} \\
2\mu \left(\frac{\partial u_n}{\partial \mathbf{t}} + \frac{\partial u_t}{\partial \mathbf{n}} \right) &= 0
\end{aligned}
\tag{4.10}
$$

Note that from equations (4.10), the dynamic conditions usually imposed in a potential flow problem (i.e. equilibrum of pressure at both sides of the boundary) can be obtained by simply setting the viscosity to zero :

$$
p_{in} = p_{out} \tag{4.11}
$$

4.3 Open boundary conditions

Numerical solutions of most of the computational fluid dynamics problems are driven by the boundary conditions, it is then extremely important to prescribe physically realistic boundary conditions that will at the same time give a well-posed mathematical problem. In numerical ship hydrodynamics, there are usually three types of boundaries: solid wall boundaries, free surface, and open boundaries. Solid wall boundary conditions are usually known (no-slip condition for viscous flow or no-penetration condition for inviscid flow). The open boundaries appear due to the truncation of the flow domain and are usually of two types: the *inflow boundary* and the *outflow boundary.*
At the inflow boundary, the conditions are usually known and easy to implement as they are imposed values. The outflow boundary conditions need some care as they are more difficult to apply since they are a priori unknown. Theoritically, the stress continuity (4.9) have to be prescribed. Usually, the outflow boundaries are located where the flow behaviour is known, for instance approximatively unidirectional or where the stresses are known. For high Reynolds number flows far from solid objects in an external flow, or for the fully developed flow out of a channel, there is no change in any of the velocity

components in the direction across the boundary and $\sigma_n = -p$ and $\sigma_t = 0$. This gives the outflow boundary condition:

$$p = p_0 \quad \text{and} \quad \frac{\partial u_n}{\partial n} = 0 \tag{4.12}$$

Gresho [27] reviews some of the open boundary conditions for incompressible flow computations and concludes that there are some theoritical concerns about the condition $\frac{\partial u_n}{\partial n} = 0$ but that it is in practise used successfully.
In ship hydrodynamics applications, the outflow boundary must also be non-reflective and able to damp out the waves. Hinatsu [32] tests and studies two types of open boundary conditions:

- an Orlanski type condition in which the Sommerfeld radiation condition is imposed on the open boundary,
- the added dissipation zone method in which the computational domain is extended with a coarse grid where fictitious damping forces are added.

In the Orlanski type of boundary condition, a physical variable ψ (usually the pressure cf. [32]) is imposed to satisfy the condition :

$$\frac{\partial \psi}{\partial t} - c\frac{\partial \psi}{\partial x} = 0 \tag{4.13}$$

where the wave phase velocity c is determined locally by use of the wave elevation around the open boundary.
In the added dissipation zone method, waves are damped by a numerical dissipation due to fictitious damping forces added to a coarse extension of the grid. A Neumann type boundary condition is prescribed at the outmost boundary and the fictituos damping forces are assumed to have the form :

$$f = -\alpha v \tag{4.14}$$

where v is the vertical velocity component and α a positive constant.

Chapter 5

Some interesting test cases

In this chapter some interesting test cases for validation purposes of the free surface method will be described.

5.1 Standing wave

Following Hinatsu [32], the first test problem will be the small amplitude standing wave problem. Figure 5.1 shows the configuration. We consider a basin of breadth b and of depth h, both equal to unity. The initial wave elevation is sinusoidal with wave height 0.01 and then, the free surface is allowed to oscillate freely. The free slip condition was imposed at all walls so that the results could be compared more easily to potential theory. The Froude number is set to one and two Reynolds number are tested: Re = 100 and Re = 10^4. The results can be found in [32].

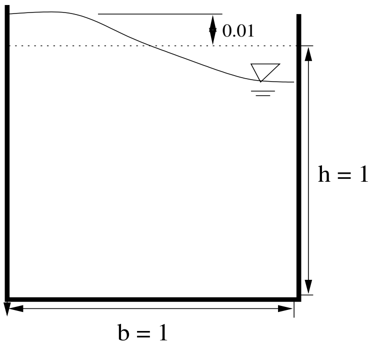


Figure 5.1: Standing wave problem

5.2 Wave generated by a bottom obstacle

The last test case is the waves generated by a bottom obstacle in a uniform flow, which also helps in assessing the performances of the open boundary conditions. The computational setup is shown figure 5.2. The shape of the obstacle is sinusoidal with a length of 4 units and a height of 0.5 unit. Reynolds number is set to 10^4 and Froude number to 1.

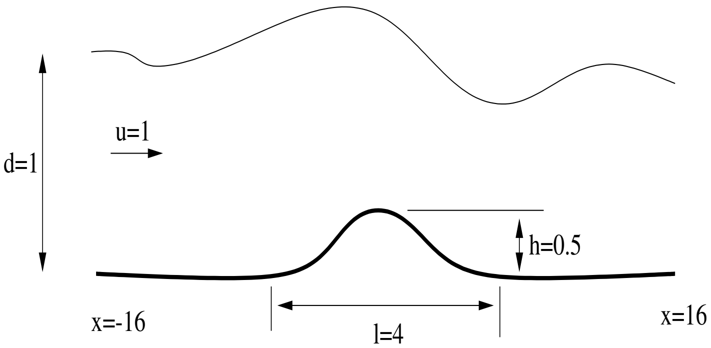


Figure 5.2: Waves generated by a bottom obstacle.

5.3 Flow around a fixed body at the free surface

Miyata et al. [54, 55] studied numerically and experimentally the flow around a two-dimensional rectangular section in deep water (figure 5.3). Results and more details about the computational setup can be found in [54, 55].

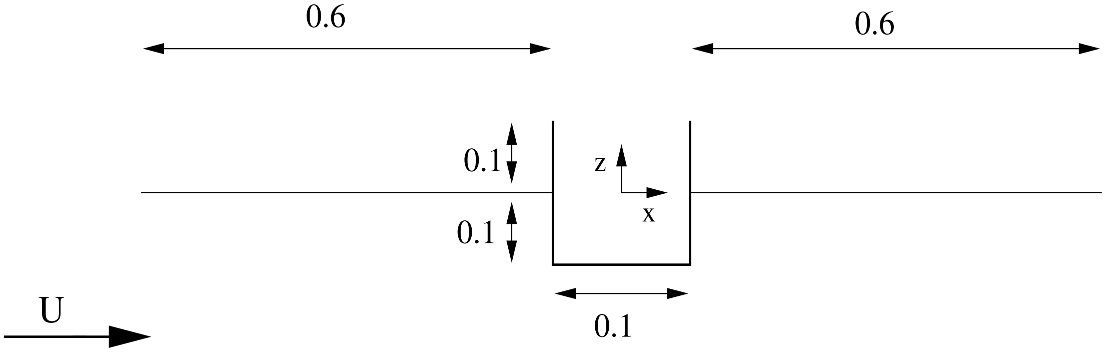


Figure 5.3: Rectangular section at the free surface.

5.4 Flow past a submerged hydrofoil

The flow past a hydrofoil under the free surface has been studied experimentally by Duncan [18] and numerically by Azcueta et al. [2] among others.
The hydrofoil with the NACA0012 profile was used. The chord length was $l_c = 203$ mm; the maximum thickness of 25.4 mm was located 61 mm behind the nose. The hydrofoil was towed at an angle of attack of 5^o with a towing velocity of 0.8 m/s. The results presented in [2] for non-breaking wave situations were for a depth of 210 mm. The Reynolds and Froude numbers based on the chord length were 162400 and 0.567 respectively (see figure 5.4).

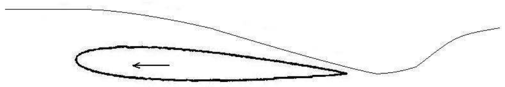


Figure 5.4: Airfoil under the free surface.

Bibliography

[1] R. Azcueta, S. Muzaferija, M. Perić and S.-D. Yoo. *Computation of Breaking Bow Waves for a Very Full Hull Ship.* Proc. Seventh International Conference on Numerical Ship Hydrodynamics, Nantes, 1999

[2] R. Azcueta, S. Muzaferija, M. Perić and S.-D. Yoo. *Computation of flow around hydrofoils under the free surface.* Proc. Seventh International Conference on Numerical Ship Hydrodynamics, Nantes, 1999

[3] K. Aziz and J.D. Hellums. *Numerical Solution of the Three-Dimensional Equations of Motions for Laminar Natural Convection.* Phys. Fluids, vol. 10, pp. 314-324, 1967

[4] D. Barkley and R. D. Henderson. *Three-dimensional Floquet stability analysis of the wake of a circular cylinder.* J. Fluid Mech., vol. 322, pp. 215-241, 1996

[5] R.F. Beck, A.M. Reed and E.P Rood. *Application of Modern Numerical Methods in Marine Hydrodynamics.* SNAME Transactions, vol. 104, pp. 519-537, 1996

[6] M. Beddhu, M. Jiang, D.L. Whitfield and L.K. Taylor. *Computation of the Wetted Transom Stern Flow over Model 5415.* Proc. Seventh International Conference on Numerical Ship Hydrodynamics, Nantes, 1999

[7] M. Beddhu, R. Pankajakshan, M. Jiang, M. Remotigue, C. Sheng, L.K. Taylor W. Briley and D.L. Whitfield. *Computation of Nonlinear Turbulent Free Surface Flows Using the Parallel Uncle Code.* Proc. 23^{rd} Symp. Naval Hydro., Val-de-Rueil, France, 2000, pp. 86-99

[8] F. Bet, D. Hanel and S. Sharma. *Numerical Simulation of Ship Flow by a Method of Artificial Compressibility.* Proc. 22^{nd} Symp. Naval Hydro., Wash. D.C., 1998, pp. 173-182

[9] F. Bet, N. Stuntz, D. Hanel and S. Sharma. *Numerical Simulation of Ship Flow in Restricted Water.* Proc. Seventh International Conference on Numerical Ship Hydrodynamics, Nantes, 1999

[10] Y.C. Chang, T.Y. Hou, B. Merriman, and S. Osher. *A Level Set Formulation of Eulerian Interface Capturing Methods for Incompressible Fluid Flows.* Journal of Computational Physics, vol. 124, pp. 449-464, 1996

[11] S. Chen, D.B. Johnson, P.E. Raad and D. Fadda. *The surface marker and micro cell method.* International Journal for Numerical Methods in Fluids, vol. 25, no. 7, pp. 749-778, 1997

[12] A. J. Chorin. *A Numerical Method for Solving Incompressible Viscous Flow Problems.* J. Comput. Phys., vol. 2, pp. 12-26, 1967

[13] A. J. Chorin. *Numerical Solution of the Navier-Stokes Equations.* Math. Comput., vol. 22, pp. 742-762, 1968

[14] H. H. Chun, I.R. Park and S. K. Lee. *Analysis of Turbulence Free-Surface Flow around Hulls in Shallow Water Channel by a Level-Set Method.* Proc. 23^{rd} Symp. Naval Hydro., Val de Rueil, France, 2000

[15] A. Cura Hochabaum and C. Schumann. *Free Surface Vsicous Flow around ship models.* Proc. 7^{th} International Conference on Numerical Ship Hydrodynamic, Nantes, 1999

[16] D. Dommermuth, G. Innis, T. Luth, E. Novikov, E. Schlageter and J. Talcott. *Numerical Simulation of Bow Waves.* Proc. 22^{nd} Symp. Naval Hydro., Wash. D.C., 1998, pp. 508-521

[17] J. Donea, S. Guiliani and J.P. Halleux. *An arbitrary Lagrangian-Eulerian Finite Element Method for Transient Dynamic Fluid-Structure Intercations.* Computer Methods in Applied Mechanics and Engineering, vol.33, pp.689-723, 1982

[18] J.H. Duncan. *The breaking and non-breaking wave resistance of a two-dimensional hydrofoil.* J. Fluid Mech., vol. 126, pp. 507-520, 1983

[19] J. Farmer, L. Martinelli and A. Jameson. *A Fast Multigrid Method for Solving the Nonlinear Ship Wave Problem with a Free Surface.* Proc. Sixth International Conference on Numerical Ship Hydrodynamics, Iowa City, 1993

[20] J. Farmer, L. Martinelli and A. Jameson. *Fast Multigrid Method for Solving Incompressible Hydrodynamic Problems with Free Surfaces.* AIAA Journal, vol. 32, pp. 1175-1182, 1994

[21] G. Fekken, A.E.P. Veldman and B. Buchner. *Simulation of Green Water Loading Using the Navier-Stokes Equations.* Proc. Seventh International Conference on Numerical Ship Hydrodynamics, Nantes, 1999

[22] J.H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics.* Springer, 3rd edition, 2002.

[23] C.A.J. Fletcher. *Computational Techniques for Fluid Dynamics - Volume II.* Springer-Verlag, Second edition, 1991

[24] L. Gentaz, P.E. Guillerm, B. Alessandrini and G. Delhommeau. *Three-Dimensional Free Surface Viscous Flow Around a Ship in Forced Motion.* Proc. Seventh International Conference on Numerical Ship Hydrodynamics, Nantes, 1999

[25] J.J Gorski. *Present state of numerical ship hydrodynamics and validation experiments.* Proceedings of OMAE'01 20th International Conference on Offshore Mechanics and Artic Engineering, Rio de Janeiro, 2001

[26] P.M. Gresho. *On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a Finite Element method that also introduces a nearly consistent mass matrix. Part 1: theory.* Int. J. Numer. Meth. Fluids vol. 11, pp. 587-620, 1990

[27] P.M. Gresho. *Incompressible Fluid Dynamics: Some Fundamental Formulation Issues.* Annu. Rev. Fluid Mech., vol. 23, pp. 413-453, 1991

[28] P.M. Gresho and R.L. Sani. *Incompressible Flow and the Finite Element Method. Advection-Diffusion and Isothermal Laminar Flow.* John Wiley and Sons, 1998

[29] J.-L. Guermond and L. Quartapelle. *On stability and convergence of projection methods based on pressure Poisson equation.* Int. J. Numer. Meth. Fluids vol. 26 pp. 1039-1053, 1998

[30] F. Harlow and Welch. *Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface.* The Physics of Fluid, vol. 8, pp. 2182-2189, 1965

[31] R.D. Henderson. *Nonlinear dynamics and pattern formation in turbulent wake transition.* J. Fluid Mech., vol. 352, pp. 65-112, 1997

[32] M. Hinatsu. *Numerical Simulation of Unsteady Viscous Nonlinear Waves Using Moving Grid System Fitted on a Free Surface.* J. Kansai Soc. N. A., Japan, No.217, 1992

[33] T. Hino. *Navier-Stokes Computations of Ship Flows on Unstructured grids.* Proc. 22nd Symp. Naval Hydro., Wash. D.C., 1998

[34] G.J. Hirasaki and J.D. Hellums. *A general formulation of the boundary conditions on the vector potential in three-dimensional hydrodynamics.* Q. Appl. Math., vol. 26, pp. 331-342, 1968

[35] C. W. Hirt and B. D. Nichols. *Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries.* Journ. Comp. Phys., vol. 39, pp. 201-225, 1981

[36] B.R Hodges and R.L. Street. *On Simulation of Turbulent Nonlinear Free-Surface Flows.* Journ. Comp. Phys., vol.151, pp.425-457, 1999

[37] J. Holmen and M. Ofstad Henriksen. *Splitting methods applied to the incompressible Navier-Stokes equations: effects on the calculation of wall-forces.* In Some numerical methods for the simulation and laminar and turbulent incompressible flows, Ph.D. thesis 2002:6, Norwegian University of Science and Technology, 2002

[38] J. Holmen and M. Ofstad Henriksen. *Splitting methods applied to the incompressible Navier-Stokes equations: effects on the calculation of wall-forces.* Submitted to Int. Jour. Numer. Meth. Fluids.

[39] T.J.R Hughes, W.K. Liu and A. Brooks. *Finite Elements Analysis of Incompressible viscous Flows by the Penalty Function Formulation.* J. Comp. Phys., vol. 30, pp. 1-60, 1979

[40] T.J.R Hughes, L.P. Franca and M. Balestra. *A new finite element formulation for computational fluid dynamic: V. Circumventing the Babuška-Brezzi condition: A stable Petrov-Galerkin formulation of the Stokes problem accomodating equal-order interpolations.* Comp. Meth. Appl. Mech. Eng., vol. 59, pp.85-99, 1986

[41] A. Iafrati, A. Olivieri, F. Pistani and E. Campana. *Numerical and Experimental Study of the Wave Breaking Generated by a Submerged Hydrofoil.* Proc. 23^{rd} Symposium on Naval Hydrodynamics, Val de Rueil, France, 2000

[42] A. Jameson, W. Schmidt and E. Turkel. *Numerical Solution of the Euler Equations by Finite Volume Methods using Runge-Kutta Time-Stepping Schemes.* AIAA-81-1259, 1981

[43] G.E. Karniadakis, M. Israeli and S.A. Orszag. *High-Order Splitting Methods for the Incompressible Navier-Stokes Equations.* Jour. Comp. Phys., vol. 97, pp. 414-443, 1991

[44] T. Kawamura and H. Miyata. *Simulation of Nonlinear Ship Flows by Density-Function Method.* Journal of The Society of Naval Architects of Japan, vol. 176, pp.1-10, 1994

[45] J. Kim and P. Moin. *Application of a Fractional-Step Method to Incompressible Navier-Stokes Equations.* Jour. Comp. Phys., vol. 59, pp. 308-323, 1985

[46] Y. Kodama (ed). *1994 Proceedings CFD Workshop Tokyo.* Ship Research Inst. Ministry of Transport, Ship and Ocean Foundation, Tokyo, Japan, Vols. 1 and 2

[47] H. Lamb. *Hydrodynamics.* Cambridge University Press, 1932

[48] H. P. Langtangen. *Computational Partial Differential Equations - Numerical Methods and Diffpack Programming.* Lecture Notes in Computational Science and Engineering vol. 2, Springer, 1991

[49] L. Larsson, B. Regnström, L. Broberg, D-Q. Li and C-E Janson. *Failures, Fantasies, and Feats in the Theoritical/Numerical Prediction of Ship Performance.* Proc. 22^{nd} Symp. Naval Hydro., Wash. D.C., 1998

[50] L. Larsson, F. Stern and V. Bertram (eds). *Gothenburg 2000, A Workshop on Numerical Ship Hydrodynamics.* Gothenburg, Sweden

[51] R.J. LeVeque. *Numerical Methods for Conservation Laws.* Lectures in Mathematics, Birkhäuser, 1992

[52] R. Löhner, C. Yang and E. Oñate. *Viscous Free Surface Hydrodynamics using unstructured grids.* Proc. 22^{nd} Symp. Naval Hydro., pp. 476-490, Wash. D.C., 1998

[53] R. Löhner. *Applied CFD Techniques - An Introduction based on Finite Element Methods.* John Wiley & Sons Ltd, 2001

[54] H. Miyata, H. Kajitani, N. Suzuki and C. Matsukawa. *Numerical and Experimental Analysis of Nonlinear Bow and Stern Waves of a Two-Dimensional Body. (First Report)* Journal of The Society of Naval Architects of Japan, vol. 154, pp. 48-55, 1983

[55] H. Miyata, H. Kajitani, C. Matsukawa, N. Suzuki, M. Kanai and S. Kuzumi. *Numerical and Experimental Analysis of Nonlinear Bow and Stern Waves of a Two-Dimensional Body. (Second Report)* Journal of The Society of Naval Architects of Japan, vol. 155, pp. 11-17, 1984

[56] H. Miyata. *Finite-difference simulation of breaking waves.* Journal of Computational Physics, vol. 65, pp. 179-214, 1986

[57] S. Muzaferija and M. Perić. *Computation of Free-Surface Flows using the Finite-Volume Method and Moving Grids.* Numerical Heat Transfer, Part B, vol. 32, pp. 396-384, 1997

[58] S. Muzaferija and M. Perić. *Computation of free surface flows using interface-tracking and interface-capturing methods.* Chap.2 in O. Mahrenholtz and M. Markiewicz (eds.), Nonlinear Water Wave Interaction, pp. 59-100, WIT Press, 1999

[59] T. Nakayama and M. Mori. *An Eulerian finite element method for time-dependent free surface problems in hydrodynamics.* International Journal for Numerical Methods in Fluids, vol. 22, pp. 175-194, 1996

[60] B.D. Nichols and C.W. Hirt. *Improved Free Surface Boundary Conditions for Numerical Incompressible-Flow Calculations.* Journal of Computational Physics, vol. 8, pp. 434-448, 1971

[61] B.D. Nichols and C.W. Hirt. *Calculating Three-Dimensional Free Surface Flows in the Vicinity of Submerged and Exposed Structures.* Journal of Computational Physics, vol. 12, pp. 234-246, 1973

[62] B.D. Nichols and C.W. Hirt. *Methods for Calculating Multi-Dimensional, transient, free surface flows past bodies.* Proceedings of the First International Conference on Numerical Ship Hydrodynamics, pp. 253-277, Gaithersburg, 1975

[63] W.F. Noh and P. Woodward. *SLIC (Simple Line Interface Calculation).* Proceedings of the 5^{th} International Conference on Numerical Methods in Fluids, 1976

[64] M. Ofstad Henriksen and J. Holmen. *Algebraic Splitting for Incompressible Navier-Stokes Equations.* Jour. Comp. Phys., vol. 175, pp. 438-453, 2002

[65] S. Osher and J.A. Sethian. *Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations.* Journal of Computational Physics, vol. 79, pp. 12-49, 1988

[66] S.V. Patankar and D.B. Spalding. *A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows.* Int. Jour. Heat Mass Transfer, vol.15, pp. 1787-1806, 1972

[67] J.B. Perot. *An analysis of the fractional-step method.* Jour. Comp. Phys., vol. 108, pp.51-58, 1993

[68] L. Quartapelle. *Numerical Solution of the Incompressible Navier-Stokes Equations.* International Series of Numerical Mathematics vol. 113, Birkäuser, 1993

[69] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations.* Springer Series in Computational Mathematics vol. 23, Springer, 1994

[70] A. Quarteroni, F. Saleri and A. Veneziani. *Analysis of the Yosida method for the incompressible Navier-Stokes equations.* Jour. Math. Pures Appl., pp. 473-503, 1999

[71] A. Quarteroni, F. Saleri and A. Veneziani. *Factorization methods for the numerical approximation of Navier-Stokes equations.* Comp. Meth. Appl. Mech. Eng., vol. 188, pp 505-526, 2000

[72] J.D. Ramshaw and J.A. Trapp. *A Numerical Technique for Low-Speed Homogeneous Two-Phase Flow with Sharp Interface.* Journal of Computational Physics, vol. 21, pp. 438-453, 1976

[73] S.G. Rubin and P.K. Khosla. *Navier-Stokes calculations with a coupled stronglu implicit method - I.* Computers and Fluids, vol. 9, pp. 163-180, 1981

[74] M. Rudman. *Volume-Tracking Methods for Interfacial Flow Calculations.* International Journal for Numerical Methods in Fluids, vol. 24, pp. 671-691, 1997

[75] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy and W. Lorensen. *Object-Oriented Modeling and Design.* Prentice Hall Inc., 1991

[76] Y. Saad. *Iterative Methods for Sparse Linear Systems.* PWS Publishing Company, 1996

[77] J. A. Sethian. *Level Set Methods and Fast Marching Methods.* Cambridge University Press, 1999

[78] M. Sussman, P. Smereka and S. Osher. *A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow.* Journal of Computational Physics, vol. 114, pp. 146-159, 1994

[79] M. Sussman, E. Fatemi, P. Smereka and S. Osher. *An Improved Level Set Method for Incompressible Two-Phase Flows.* Computers and Fluid, vol. 27, no. 5, pp. 663-680, 1998

[80] M. Sussman and D. Dommermuth. *The Numerical Simulation of Ship Waves Using Cartesian Grid Methods.* Proc. 23^{rd} Symp. Naval Hydro., Val de Rueil, France, 2000

[81] J. C. Tannehill, D. A. Anderson and R. H. Pletcher. *Computational Fluid Mechanics.* Series in computational and physical processes in mechanics and thermal sciences, Second edition, 1997

[82] R. Temam. *Sur l'approximation de la solution des é quations de Navier-Stokes par la méthode des pas fractionnaires (I).* Arch. Ration. Mech. Anal., vol. 32, pp. 135-153, 1969

[83] H. Tennekes and J.L. Lumley. *A First Course in Turbulence.* The MIT Press, 1972

[84] T.G. Thomas, J.J.R. Williams and D.C. Leslie. *Development of a conservative 3D free surface code.* Journal of Hydraulics Research, vol. 30, no.1, pp. 107-115, 1992

[85] T.G. Thomas, D.C. Leslie and J.J.R. Williams. *Free surface simulations using a conservative 3D code.* Journal of Computational Physics, vol. 116, pp. 52-68, 1995

[86] M. Tomé and S. McKee. *GENSMAC : A computational Marker-And-Cell method for free surface flows in general domains.* Journal of Computational Physics, vol. 110, pp. 171-186, 1994

[87] S. Turek. *Efficient Solvers for Incompressible Flow Problems.* Lecture Notes in Computational Science and Engineering vol. 6, Springer, 1999

[88] B. Vallès. *Computational study of vortex shedding behind bluff bodies.* Ph.D. thesis, Norwegian University of Science and Technology, 2001

[89] E.F. Van de Velde. *Concurrent Scientific Computing.* Springer-Verlag, 1994

[90] J. Van Kan. *A second-order accurate pressure-correction scheme for viscous incompressible flow.* SIAM J. Sci. Stat Comput, vol. 7, No. 3, pp. 870-891, 1986

[91] M. Vogt and L. Larsson. *Level Set Methods for Predicting Viscous Free Surface Flows.* Proc. Seventh International Conference on Numerical Ship Hydrodynamics, Nantes, 1999

[92] J.F. Wendt. *Computational Fluid Dynamics - An Introduction.* A von Karman Institue Book, Springer, 1996

[93] F. M. White. *Fluid Mechanics.* McGraw-Hill International Editions, 4^{th} edition, 1999

[94] D. Youngs. *Time-Dependent Multi-Material Flow with Large Fluid Distortion.* In Numerical Methods for Fluid Dynamics, K. Morton and M. Baines (eds.), Academic Press, pp. 273-285, 1982

[95] S. Zalesak. *Fully Multi-Dimensional Flux-Corrected Transport Algorithms for Fluids.* Journal of Computational Physics, vol. 11, pp. 38-69, 1979

[96] O.C. Zienkiewicz and R. Taylor. *The Finite Element Method.* McGraw-Hill, London, 1989

Index