ODDLS: A new unstructured mesh finite element method for the analysis of free surface flow problems

Julio Garcia-Espinosa[1, 2], Aleix Valls[1, 2] and Eugenio Oñate[2, *, †]

[1] *COMPASS Ingeniería y Sistemas, S.A., Tuset 8, 7-2, 08006 Barcelona, Spain*
[2] *International Center for Numerical Methods in Engineering (CIMNE), Universidad Politécnica de Cataluña, Gran Capitán s/n, 08034 Barcelona, Spain*

SUMMARY

This paper introduces a new stabilized finite element method based on the finite calculus (*Comput. Methods Appl. Mech. Eng.* 1998; **151**:233–267) and arbitrary Lagrangian–Eulerian techniques (*Comput. Methods Appl. Mech. Eng.* 1998; **155**:235–249) for the solution to free surface problems. The main innovation of this method is the application of an overlapping domain decomposition concept in the statement of the problem. The aim is to increase the accuracy in the capture of the free surface as well as in the resolution of the governing equations in the interface between the two fluids. Free surface capturing is based on the solution to a level set equation. The Navier–Stokes equations are solved using an iterative monolithic predictor–corrector algorithm (*Encyclopedia of Computational Mechanics*. Wiley: New York, 2004), where the correction step is based on imposing the divergence-free condition in the velocity field by means of the solution to a scalar equation for the pressure. Examples of application of the ODDLS formulation (for overlapping domain decomposition level set) to the analysis of different free surface flow problems are presented.

KEY WORDS: finite element method; free surface; flow problems

INTRODUCTION

The prediction of the free surface motion of liquids is a topic of big relevance in many engineering fields. Despite recent advances in computational fluid dynamics, the development of an efficient,

[*] Correspondence to: Eugenio Oñate, International Center for Numerical Methods in Engineering (CIMNE), Universidad Politécnica de Cataluña, Gran Capitán s/n, 08034 Barcelona, Spain.

[†] E-mail: onate@cimne.upc.edu

Contract/grant sponsor: Ministerio de Educación y Ciencia, Spain; contract/grant numbers: TRA2005-07536 07536/ TMAR, VEM2004-08641-C03-03

accurate and robust numerical algorithm for the analysis of problems with large free surface deformation is still a challenging issue.

The free surface flow problem can be considered as a particular case of the more general problem of predicting the interface between two immiscible fluids: the flowing liquid (typically water) and air. Computing the interface between two immiscible fluids is difficult because neither the shape nor the position of the interface between the two fluids is known *a priori*. There are basically two approaches for computing free surfaces in this kind of flows: *interface-tracking* and *interface-capturing* methods. The former computes the motion of the flow particles based on a Lagrangian approach, where the numerical domain adapts itself to the shape and position of the free surface. Different numerical techniques, such as the smoothed particle hydrodynamics (SPH) method [1, 2] and the particle finite element method (PFEM), belong to this kind [3–5]. In interface-tracking methods, the free surface is treated as a boundary of the computational domain where the kinematic and dynamic boundary conditions are applied. The main problems of this approach are the large computational effort required due to the need of updating the analysis domain every time step and the difficulty in imposing mass continuity in an accurate way.

Standard interface-capturing methods consider both fluids as a single effective fluid with variable properties [6–11]. The interface is considered as a region of sudden change in the fluid properties. This approach requires an accurate modelling of the jump in the properties of the two fluids taking into account that the free surface can move, bend and reconnect in arbitrary ways. Furthermore, the imposition of the exact boundary conditions in the interface is usually simplified.

This paper shows a new stabilized FEM for solving the Navier–Stokes equations including free surface effects, which overcomes most of the difficulties of the existing methodologies. The starting point is the modified governing differential equations for an incompressible viscous flow and the free surface condition, incorporating stabilization terms via a finite calculus (FIC) procedure [12–18]. The main innovation of the new method termed overlapping domain decomposition level set (ODDLS) is the introduction of an *overlapping domain decomposition* concept in the statement of the problem for increasing the accuracy in the capture of free surface as well as in the resolution of the governing equations in the interface between the two fluids. Free surface capturing is based on the solution to a *level set-type equation* [6–8], whereas the solution to the Navier–Stokes equations is based on an implicit monolithic second-order method originally proposed by Soto *et al*. [19]. This scheme is derived by splitting the momentum equation in a similar manner as in an implicit fractional step method [13, 14, 17, 19].

The outline of this paper is as follows. In the following section, the statement of the governing equations of two incompressible and immiscible fluids is presented. Then, the treatment of the interfacial boundary condition is analysed and the FIC-stabilized problem is presented. The overlapping domain decomposition methodology is then applied to the stabilized problem. The discretization of the FIC-governing equations using equal-order linear finite element is described and finally, the arbitrary Lagrangian–Eulerian (ALE) version of the method is introduced. The accuracy of the new formulation for the analysis of free surface flows is verified in different validation cases.

STATEMENT OF THE PROBLEM

The velocity and pressure fields of two incompressible and immiscible fluids moving in the domain $\Omega \subset \mathbb{R}^d$ $(d=2,3)$ during the time interval $(0,T]$ can be described by the incompressible

Navier–Stokes equations for multiphase flows, also known as the non-homogeneous incompressible Navier–Stokes equations [20]:

$$\begin{aligned} \partial_t \rho + \nabla(\rho \mathbf{u}) &= 0 \\ \partial_t(\rho \mathbf{u}) + \nabla\cdot(\rho \mathbf{u}\otimes\mathbf{u}) - \nabla\cdot\boldsymbol{\sigma} &= \rho \mathbf{f} \\ \nabla\cdot\mathbf{u} &= 0 \end{aligned} \tag{1}$$

where ρ is the fluid density field, $\mathbf{u}$ is the velocity field and $\boldsymbol{\sigma}$ is the Cauchy stress tensor defined as

$$\begin{aligned} \boldsymbol{\sigma} &= -p\mathbf{I} + \boldsymbol{\tau} \\ \boldsymbol{\tau} &= \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}) \end{aligned} \tag{2}$$

where μ is the dynamic viscosity and $\mathbf{I}$ is the identity matrix. Boldface characters are used to denote vector and tensor variables.

As we consider problems with a moving interface inside the domain, all subdomains and functions defined hereafter are time dependent.

For $\forall t \in [0, T)$, let $\Omega_1(t) = \{\mathbf{x}\in\Omega | \mathbf{x} \text{ is Fluid}_1\}$ be the part of the domain Ω occupied by fluid 1 and let $\Omega_2(t) = \{\mathbf{x}\in\Omega | \mathbf{x} \text{ is Fluid}_2\}$ be the part of the domain Ω occupied by fluid 2. Therefore, $\Omega_1(t)$ and $\Omega_2(t)$ are two disjoint subdomains of Ω. Then

$$\Omega = \mathrm{int}(\overline{\Omega_1(t) \cap \Omega_2(t)}) \quad \forall t \in (0, T] \tag{3}$$

where 'int' denotes the topological interior and the over-bar indicates the topological adherence of a given set, for more details see [21]. The system of equations (1) must be completed with the necessary initial and boundary conditions, as shown below.

It is usual in the literature to consider that the first equation of system (1) is equivalent to impose a divergence-free velocity field (the third equation in (1)), as the density is taken as a constant. However, for multiphase incompressible flows, the density cannot be considered to be constant in $\Omega\times(0, T]$. In fact, it is possible to define the ρ, μ fields as follows:

$$\rho(\mathbf{x}, t), \mu(\mathbf{x}, t) = \begin{cases} \rho_1, \mu_1, & \mathbf{x}\in\Omega_1(t) \\ \rho_2, \mu_2, & \mathbf{x}\in\Omega_2(t) \end{cases} \quad \forall(\mathbf{x}, t)\in\Omega\times(0, T] \tag{4}$$

Let $\psi : \Omega\times(0, T] \to \mathbb{R}$ be a function named the *level set* function hereafter and defined as follows:

$$\psi(\mathbf{x}, t) = \begin{cases} d(\mathbf{x}, \Gamma(t)), & \mathbf{x}\in\Omega_1(t) \\ 0, & \mathbf{x}\in\Gamma(t) \\ -d(\mathbf{x}, \Gamma(t)), & \mathbf{x}\in\Omega_2(t) \end{cases} \tag{5}$$

where $d(\mathbf{x}, \Gamma(t)) := \inf\{\|\mathbf{x}-\mathbf{y}\| | \mathbf{y}\in\Gamma(t)\}$ is the distance of point $\mathbf{x}$ to the interface between the two fluids, denoted by $\Gamma(t)$, at time t.

From definition (5), it is trivially obtained that

$$\Gamma(t) = \{\mathbf{x}\in\Omega | \psi(x, t) = 0\} \tag{6}$$

Therefore, it is possible to re-write definition (4) as

$$\rho, \mu = \begin{cases} \rho_1, \mu_1, & \psi > 0 \\ \rho_2, \mu_2, & \psi < 0 \end{cases} \tag{7}$$

Let us express the density fields in terms of the level set function ψ as

$$\rho(\mathbf{x}, t) = \rho(\psi(\mathbf{x}, t)) \quad \forall (\mathbf{x}, t) \in \Omega \times (0, T] \tag{8}$$

Then, the density derivatives can then be expressed as

$$\partial_t \rho = d_\psi \rho \cdot \partial_t \psi, \quad \nabla \rho = d_\psi \rho \cdot \nabla \psi \tag{9}$$

Substituting relation (9) into the first equation of system (1) gives

$$\begin{aligned} \partial_t \rho + \nabla(\rho \mathbf{u}) \underset{\nabla \cdot \mathbf{u} = 0}{=} \partial_t \rho + (\mathbf{u} \cdot \nabla) \rho &= d_\psi \rho \cdot \partial_t \psi + d_\psi \rho \cdot (\mathbf{u} \cdot \nabla) \psi \\ &= d_\psi \rho [\partial_t \psi + (\mathbf{u} \cdot \nabla) \psi] = 0 \end{aligned} \tag{10}$$

The multiphase Navier–Stokes problem (1) is therefore equivalent to solving the following system of equations:

$$\begin{aligned} \partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) - \nabla \cdot \boldsymbol{\sigma} &= \rho \mathbf{f} \\ \nabla \cdot \mathbf{u} &= 0 \end{aligned} \tag{11}$$

coupled with the equation

$$\partial_t \psi + (\mathbf{u} \cdot \nabla) \psi = 0 \tag{12}$$

Equation (12) defines the transport of the level set function due to the velocity field obtained by solving Equations (11).

As a conclusion, the free surface capturing problem can be described by Equations (11) and (12). The interface between the two fluids is defined by the level set 0 of ψ.

It is possible to prove, assuming the variables of the problem as sufficiently smooth, that Equations (1), or equivalently the system given by Equations (11) and (12), have a unique global solution [21].

Denoting by an over-bar the prescribed values and by $\mathbf{n}$ the normal to the boundary, the boundary conditions of Equations (11) and (12) are

$$\begin{aligned} \mathbf{u} &= \bar{\mathbf{u}} \quad \text{on } \Gamma_{\mathrm{D}} \\ \mathbf{n} \cdot \boldsymbol{\sigma} &= \bar{\mathbf{t}} \quad \text{on } \Gamma_{\mathrm{N}} \\ \mathbf{n} \cdot \mathbf{u} = u_n, \quad & \left. \begin{aligned} \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{g} &= \bar{t}_1 \\ \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{s} &= \bar{t}_2 \end{aligned} \right\} \quad \text{on } \Gamma_{\mathrm{M}} \end{aligned} \tag{13}$$

where the boundary $\partial\Omega$ of the domain Ω has been split into three disjoint sets: $\Gamma_{\mathrm{D}}, \Gamma_{\mathrm{N}}$, where the Dirichlet and Neumann boundary conditions are imposed, and Γ_{M} where mixed conditions are imposed. Mixed Dirichlet–Neumann boundary conditions are usually applied when wall functions are used for modeling the behaviour of the flow close to solid walls. Vectors $\mathbf{g}, \mathbf{s}$ in the above equation span the space tangent to Γ_{M}. Owing to the fact that (12) is an hyperbolic equation, only boundary conditions must be imposed at the inlet boundary noted by Γ_{inlet} and defined as

$$\Gamma_{\text{inlet}}=\{\mathbf{x}\in\Gamma|\mathbf{n}(\mathbf{x})\cdot\mathbf{u}(\mathbf{x})<0\} \tag{14}$$

Then the boundary condition for Equation (12) is

$$\psi=\bar{\psi} \quad \text{on } \Gamma_{\text{inlet}} \tag{15}$$

Finally, the initial conditions for the problem are

$$\mathbf{u}=\mathbf{u}_0 \text{ in } \Omega, \quad \psi=\psi_0 \text{ in } \Omega \tag{16}$$

Remark 1

The initial condition for ψ can be directly derived from the initial position of the interface. Using the definition of ψ given in Equation (5), ψ_0 can be computed as the signed distance to the initial interface position, $\Gamma_0=\Gamma(0)=\{x\in\Omega|\psi(x,0)=0\}$. An efficient algorithm to calculate the signed distance to the interface is presented next.

HOW TO CALCULATE A SIGNED DISTANCE TO THE INTERFACE

Based on the flow properties, such as density or viscosity, we can calculate an initial guess for ψ as follows:

$$\psi(\mathbf{x},t)=\begin{cases}1 & \text{if } \rho(\mathbf{x},t)=\rho_1\\ 0, & \mathbf{x}\in\Gamma(t)\\ -1 & \text{if } \rho(\mathbf{x},t)=\rho_2\end{cases} \tag{17}$$

Clearly, the level set function obtained from Equation (17) is not a signed distance to Γ. We need to *reinitiate* ψ in order to guarantee that ψ fulfils definition (5). We use a technique based on the following property to reinitiate the level set function as a signed distance to Γ [9]:

If ψ is a signed distance function to Γ, then

$$\|\nabla\psi\|=1 \quad \forall(\mathbf{x},t)\in\Omega\times(0,T] \tag{18}$$

where $\|\cdot\|$ denotes the Euclidian norm in $\mathbb{R}^d$.

We use Equation (18) to recalculate ψ. In case Equation (18) is not satisfied, then

$$1-\|\nabla\psi\|=r \tag{19}$$

The residual r in Equation (19) can be understood as the time derivate of ψ with respect to a pseudo-time τ, i.e.

$$r:=\partial_\tau\psi \tag{20}$$

Then, the stationary solutions to Equation (20) (i.e. when $\partial_\tau \psi = 0$) satisfy Equation (18). In summary, for a given time $t \in [0, T]$, we calculate ψ as the stationary solution to the following hyperbolic problem:

$$\begin{aligned} \partial_\tau \psi + \|\nabla \psi\| &= 1 \quad \text{on } \Omega \\ \psi(\tau = 0) &= \psi(\cdot, t) \\ \psi(\mathbf{x}, \tau) &= 0 \quad \forall \mathbf{x} \in \Gamma(t) \end{aligned} \tag{21}$$

Problem (21) can be rewritten in a more convenient manner as

$$\partial_\tau \psi + (\mathbf{w} \cdot \nabla)\psi = \operatorname{sign}(\psi), \quad \mathbf{w} = \operatorname{sign}(\psi) \frac{\nabla \psi}{\|\nabla \psi\|} \tag{22}$$

As the level set values identify the free surface between the two fluids, the following relations can be obtained:

$$\mathbf{n}(\mathbf{x}, t) = \nabla \psi|_{(\mathbf{x}, t)}, \quad \kappa(\mathbf{x}, t) = \nabla \cdot (\mathbf{n}(\mathbf{x}, t)) \tag{23}$$

where $\mathbf{n}$ is the normal vector to the interface Γ, oriented from fluid 2 to fluid 1, and κ is the local curvature of the interface.

From Equation (22) it can be easily shown that the reinitialization process begins from the interface and it propagates to the whole domain. This property is attractive from the computational point of view, as we are interested only in the accurate values of ψ in the region close to the interface. Then we only need to reach the steady-state solution to Equation (22) in a narrow band close to the interface as proposed in [7]. The reinitialization algorithm used in this work has been optimized using the concept of nodal levels described next.

Given a level set distribution ψ, for every time step t one can assign a level l_i to the node $\mathbf{x}_i$ based on the following rules:

A nodal point i of coordinates $\mathbf{x}_i$ belongs to the level $l_i = 1$, if $\psi(\mathbf{x}_i, t) \geqslant 0$, and it is connected to at least one node j with $\psi(\mathbf{x}_j, t) < 0$.

A nodal point i of coordinates $\mathbf{x}_i$ belongs to the level $l_i = -1$, if $\psi(\mathbf{x}_i, t) < 0$, and it is connected to at least one node j with $\psi(\mathbf{x}_j, t) \geqslant 0$.

A nodal point i of coordinates $\mathbf{x}_i$ belongs to the level $l_i > 1$, if $\psi(\mathbf{x}_i, t) > 0$, and it is connected to at least one node of level $l_i - 1$.

A nodal point i of coordinates $\mathbf{x}_i$ belongs to the level $l_i < -1$, if $\psi(\mathbf{x}_i, t) < 0$, and it is connected to at least one node of level $l_i + 1$.

Once the levels have been assigned to the mesh points, the reinitialization is performed in a narrow band of n levels around the interface. Equation (22) is solved using a forward Euler scheme until the steady state in the n levels is achieved. k iterations with $k > n$ are performed in such a way that for every iteration $i = 1, \ldots, k$. The iterations are carried out only for those nodes j of the mesh with $|l_j| \leqslant i$.

Owing to the evolution of the interface defined by Equation (12), the level set function does not fulfil Equation (18) after some time. It is therefore recommended to apply the reinitialization scheme described above at every time step.

INTERFACIAL BOUNDARY CONDITIONS

At a moving interface the jump condition applies. For immiscible fluids we can express the jump condition at the free surface as

$$\left.\begin{aligned}[\mathbf{n}\cdot(\boldsymbol{\sigma}\cdot\mathbf{n})]=\gamma\kappa \text{ on } \Gamma\\ [\mathbf{s}\cdot(\boldsymbol{\sigma}\cdot\mathbf{n})]=0 \text{ on } \Gamma\end{aligned}\right\} \tag{24}$$

where γ is the coefficient of surface tension (a constant of the problem), $\mathbf{s}$ is any unit vector tangent to the interface and $[\cdot]$ defines the jump across the interface:

$$[A]=A_1-A_2 \tag{25}$$

Taking into account that $\boldsymbol{\sigma}_i=-p\mathbf{I}+\mu_i(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})$, $i=1,2$, and using that $\mathbf{s}\cdot\mathbf{n}=0$, the second equation in (24) yields

$$\mathbf{s}\cdot[(\mu_1-\mu_2)(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})\cdot\mathbf{n}]=0 \tag{26}$$

From Equation (26) we can conclude that $(\mu_1-\mu_2)(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})\cdot\mathbf{n}\in\langle\mathbf{s}\rangle^{\perp}$, where $\langle\mathbf{v}\rangle^{\perp}$ denotes the orthogonal subspace to $\mathbf{v}$. Then it is clear that

$$(\mu_1-\mu_2)(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})\cdot\mathbf{n}=\lambda\mathbf{n},\quad \lambda\in\mathrm{Spec}(\boldsymbol{\tau}_1-\boldsymbol{\tau}_2) \tag{27}$$

Integrating over any closed surface ∂M containing Γ, in both sides of Equation (27) yields

$$\int_{\partial M}(\mu_1-\mu_2)\mathbf{n}\cdot(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})=\int_{\partial M}\lambda\mathbf{n} \tag{28}$$

Applying the Gauss theorem to the right-hand side of Equation (28), we can deduce that $\lambda=0$ or equivalently

$$\mathbf{n}\cdot(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})\quad \text{on } \Gamma \tag{29}$$

Introducing Equation (29) into the first condition in Equation (24), the following equivalent condition is obtained:

$$p_1\mathbf{n}=p_2\mathbf{n}+\gamma\kappa\mathbf{n} \tag{30}$$

This form of the jump condition is very useful in the development of the method presented in the following sections.

In the classical level set formulation for multiphase flow problems [9], jump condition (24) can be introduced into the momentum equation by adding an additional source term, i.e.

$$\begin{aligned}\partial_t(\rho\mathbf{u})+\nabla(\rho\mathbf{u}\otimes\mathbf{u})-\nabla\cdot\boldsymbol{\sigma}&=\rho\tilde{\mathbf{f}}\\ \tilde{\mathbf{f}}&=\mathbf{f}+\gamma\kappa\mathbf{n}\delta(\Gamma)\end{aligned} \tag{31}$$

where $\delta(\Gamma)$ is Dirac's delta function on Γ. As the properties of the fluids change discontinuously across the interface, the direct solution to Equation (31) has important numerical drawbacks: inaccurate resolution of the pressure at the interface, fluid mass losses, inaccurate interface location, etc.

A common approach to overcome these difficulties is to smooth the discontinuous transition of the flow properties and Dirac's delta functions across the interface using of a smoothed Heaviside function. This function is usually expressed as

$$H_\varepsilon(\psi)=\begin{cases}0, & \psi<-\varepsilon\\ \dfrac{1}{2}\left[1+\dfrac{\psi}{\varepsilon}+\dfrac{1}{\pi}\sin\left(\dfrac{\pi\psi}{\varepsilon}\right)\right], & |\psi|\leqslant\varepsilon\\ 1, & \psi>\varepsilon\end{cases} \tag{32}$$

where 2ε is the thickness of the transition area. Then a property η is approximated as

$$\eta(\phi)=\eta_1+(\eta_1-\eta_2)H_\varepsilon(\phi) \tag{33}$$

and Dirac's delta function is approximated as

$$\delta(\Gamma)=d_\psi H_\varepsilon \tag{34}$$

This approach also presents several deficiencies: the accuracy of the results depends on the *ad hoc* thickness of the transition area 2ε; the jump condition is not properly captured; the interface propagation velocity is not correctly calculated; etc.

Other schemes, such as the ghost fluid method [10] or Lagrangian methods such as the PFEM [4], attempt to overcome the problem of smoothing of properties at the interface region. However, these methods have several drawbacks, such as limitations of conservative and convergence properties in some cases or an increase in the computational effort. As a conclusion, a method able to deal with a discontinuous transition of properties across the interface, the jump interface condition and computationally affordable for large models is required.

The overlapping domain decomposition method described in the following sections has been developed to overcome these difficulties.

STABILIZED FIC-FEM FORMULATION

It is well known that the standard Galerkin approximation of the incompressible Navier–Stokes equations may suffer from numerical instabilities from two main sources. The first is due to the advective character of the equations, which induces oscillations for high values of the velocity. The second source has to do with the mixed character of the equations, which limits the stability of the solution to the satisfaction of the well-known inf–sup condition. The stabilization technique chosen in this work to overcome both problems is based on the FIC method presented in [12].

The stabilized FIC form of the governing differential equations (11) and (12) is expressed as

$$\mathbf{r}_m\underline{-\tfrac{1}{2}(\mathbf{h}_m\cdot\nabla)\mathbf{r}_m}=\mathbf{0},\quad r_d\underline{-\tfrac{1}{2}(\mathbf{h}_d\cdot\nabla)r_d}=0,\quad r_\psi\underline{-\tfrac{1}{2}(\mathbf{h}_\psi\cdot\nabla)r_\psi}=0 \tag{35}$$

The boundary conditions for the stabilized FIC problem are expressed as

$$\begin{aligned} \mathbf{u} &= \bar{\mathbf{u}} \quad \text{on } \Gamma_{\mathrm{D}} \\ \mathbf{n}\cdot\boldsymbol{\sigma}\underline{-\tfrac{1}{2}\mathbf{n}\cdot\mathbf{h}_m\cdot\mathbf{r}_m} &= \bar{\mathbf{t}} \quad \text{on } \Gamma_{\mathrm{N}} \\ \mathbf{n}\cdot\mathbf{u}=\bar{u}_n, \quad \mathbf{n}\cdot\boldsymbol{\sigma}\cdot\mathbf{g}\underline{-\tfrac{1}{2}\mathbf{n}\cdot\mathbf{h}_m\cdot\mathbf{r}_m\cdot\mathbf{g}} &= \bar{t}_1 \quad \text{on } \Gamma_{\mathrm{M}} \\ \mathbf{n}\cdot\boldsymbol{\sigma}\cdot\mathbf{s}\underline{-\tfrac{1}{2}\mathbf{n}\cdot\mathbf{h}_m\cdot\mathbf{r}_m\cdot\mathbf{s}}=\bar{t}_2, \quad \psi &= \bar{\psi} \quad \text{on } \Gamma_{\mathrm{inlet}} \end{aligned} \tag{36}$$

The underlined terms in Equations (35) and (36) introduce the necessary stabilization for the numerical solution to the Navier–Stokes problem [13, 15, 17].

Note that the terms $\mathbf{r}_m, r_d$ and r_ψ denote the residuals of Equations (11) and (12), i.e.

$$\begin{aligned} \mathbf{r}_m &:= \partial_t(\rho\mathbf{u})+\nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u})+\nabla p-\nabla\cdot\boldsymbol{\sigma}-\rho\mathbf{f} \\ r_d &= \nabla\cdot\mathbf{u} \\ r_\psi &:= \partial_t\psi+(\mathbf{u}\cdot\nabla)\psi \end{aligned} \tag{37}$$

The *characteristic length* vectors $\mathbf{h}_m, \mathbf{h}_d$ and $\mathbf{h}_\psi$ contain the dimensions of the finite space domain where the balance laws of mechanics are enforced. At the discrete level these dimensions are of the order of the element or grid dimension used for the computation. Details on obtaining the FIC-stabilized equations and the recommendation for computing the characteristic length vectors can be found in [15–17].

OVERLAPPING DOMAIN DECOMPOSITION METHOD

To overcome the drawbacks related to the discontinuity of fluid properties across the interface and to impose the interfacial boundary condition we split the domain Ω into two overlapping subdomains. Based on this subdivision of the domain, we apply a Dirichlet–Neumann overlapping domain decomposition technique with appropriate boundary conditions in order to satisfy the interfacial condition.

Let $K=\bigcup_{e=1}^{\#K}(N^e, K^e, \Theta^e)$ be a finite element partition of domain Ω, where N^e are the element nodes, K^e is the element spatial domain, Θ^e are the element shape functions and K is the total number of elements in the partition.

We assume that K satisfies the following approximation property:

$$\operatorname{dist}\left(\partial\Omega, \partial\left(\bigcup_{e=1}^{\#K} K^e\right)\right) \leqslant \max_{1\leqslant e\leqslant \#K}\{\operatorname{diam}(K^e)\} \tag{38}$$

for a fixed instant $t, t\in(0,T]$. Let us consider a domain decomposition of domain Ω into three disjoint subdomains $\Omega_3(t)$, $\Omega_4(t)$ and $\Omega_5(t)$ in such a way that $\Omega_3(t)=\bigcup_e K_3^e$ and $\Omega_5=\bigcup_e K_5^e$, where K_e^3 are the elements of the finite element partition K, such as $\forall\mathbf{x}\in K_3^e|\psi(x,t)>0$, and K_e^5 are the elements of the finite element partition, such as $\forall\mathbf{x}\in K_5^e|\psi(x,t)<0$. The geometrical domain decomposition is completed with

$$\Omega_4(t)=\Omega\backslash(\Omega_3(t)\cup\Omega_5(t)) \tag{39}$$

From this partition, we define the following two overlapping domains:

$$\tilde{\Omega}_1(t) := \mathrm{int}(\overline{\Omega_3(t) \cup \Omega_4(t)}), \quad \tilde{\Omega}_2(t) := \mathrm{int}(\overline{\Omega_4(t) \cup \Omega_5(t)}) \tag{40}$$

Some comments about the $\tilde{\Omega}_1 - \tilde{\Omega}_2$ partition are given next.

In order to simplify the notation, we will omit the time dependency of domains in the remainder of this section.

Let $\partial\tilde{\Omega}_i$, $i = 1, 2$, be the boundary of $\tilde{\Omega}_i$, then

$$\partial\tilde{\Omega}_i \cap \partial\Omega = \underbrace{(\partial\tilde{\Omega}_i \cap \Gamma_{\mathrm{D}})}_{\Gamma_{iD}} \cup \underbrace{(\partial\tilde{\Omega}_i \cap \Gamma_{\mathrm{N}})}_{\Gamma_{iN}} \cup \underbrace{(\partial\tilde{\Omega}_i \cap \Gamma_{\mathrm{M}})}_{\Gamma_{iM}} \tag{41}$$

It has to be noted that $\partial\tilde{\Omega}_i$ contains an additional term $\tilde{\Gamma}_i$ that is not included into $\partial\Omega$. This term comes from the presence of an interface inside Ω. As we are using a capturing technique, the position of the interface will not lie on mesh nodes, as usually happens in a tracking technique. Therefore, some elements can be intersected by the interface. Then $\tilde{\Gamma}_i$ is defined as follows:

$$\tilde{\Gamma}_i = \partial\Omega_4 \cap \tilde{\Omega}_{i+(-1)^{i-1}} \tag{42}$$

Note that $\tilde{\Gamma}_i$ is not coincident with Γ, but the following condition is satisfied:

$$\mathrm{dist}(\Gamma, \tilde{\Gamma}_i) \leqslant \max\{\mathrm{diam}(K^e) | K^e \subset \Omega_4\} \tag{43}$$

Summarizing, we have

$$\partial\tilde{\Omega}_i = \Gamma_{iD} \cup \Gamma_{iN} \cup \Gamma_{iM} \cup \tilde{\Gamma}_i \tag{44}$$

Figure 1 is a sketch of the domain partition described above.

We have two restrictions for the definition of the boundary condition on $\tilde{\Gamma}_i$: fluid velocities must be compatible at the interface and the jump boundary condition (24) must be satisfied on Γ. The domain decomposition technique chosen is of Dirichlet–Neumann type and it allows us to fulfil both restrictions.

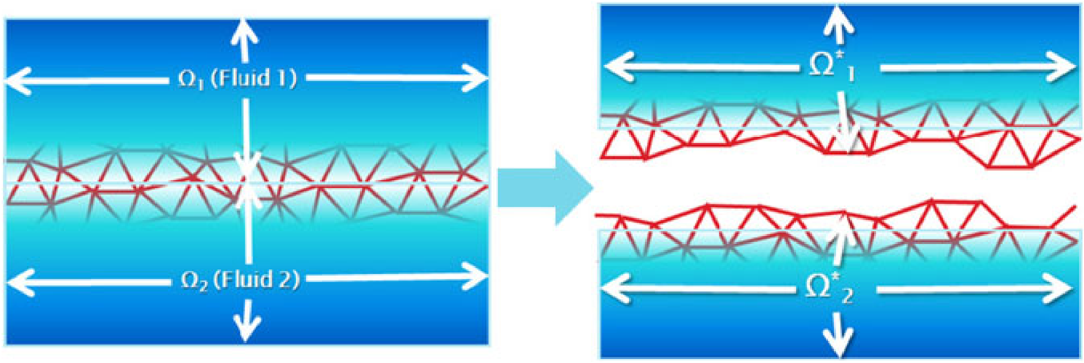

Figure 1. Geometrical domain decomposition.

Let us introduce a standard notation. We denote by x_i a variable related to $\tilde{\Omega}_i$. For example, $\mathbf{u}_1$ is the velocity field solution to problem (36), where the domain Ω has been replaced by $\tilde{\Omega}_1$.

We apply the Dirichlet conditions on $\tilde{\Gamma}_1$ and the Neumann conditions on $\tilde{\Gamma}_2$. For the boundary $\tilde{\Gamma}_1$, we use the compatibility of velocities at the interface:

$$\mathbf{u}_1 = \mathbf{u}_2 \quad \text{on } \tilde{\Gamma}_1 \tag{45}$$

For the boundary $\tilde{\Gamma}_2$, we use the jump boundary condition. We are looking for a traction vector $\tilde{\mathbf{t}}$ such that

$$\mathbf{n}_2 \cdot \boldsymbol{\sigma}_2 = \tilde{\mathbf{t}} \quad \text{on } \tilde{\Gamma}_2 \tag{46}$$

and $\tilde{\mathbf{t}}$ must guarantee that the jump boundary condition (31) holds, that is to say

$$p_1 \mathbf{n} = p_2 \mathbf{n} + \gamma \kappa \mathbf{n} \quad \text{on } \Gamma \tag{47}$$

Let us write the variational problem considering the domain decomposition above described.

Domain 1

$$\begin{aligned} &(\rho_1 \partial_t \mathbf{u}_1, \mathbf{v})_{\tilde{\Omega}_1} + \langle \rho_1 (\mathbf{u}_1 \cdot \nabla) \mathbf{u}_1, \mathbf{v} \rangle_{\tilde{\Omega}_1} + (\boldsymbol{\sigma}_1, \nabla \mathbf{v})_{\tilde{\Omega}_1} + \tfrac{1}{2} (\mathbf{r}_{1m}, (\mathbf{h}_{1m} \cdot \nabla) \mathbf{v})_{\tilde{\Omega}_1} \\ &\quad + (\bar{\mathbf{t}}, \mathbf{v})_{\Gamma_{1N}} + (\bar{t}_1 \mathbf{g} + \bar{t}_2 \mathbf{s}, \mathbf{v})_{\Gamma_{1M}} = \langle \rho_1 \mathbf{f}, \mathbf{v} \rangle_{\tilde{\Omega}_1} \\ &(q, \nabla \cdot \mathbf{u}_1)_{\tilde{\Omega}_1} + \tfrac{1}{2} (r_{1d}, (\mathbf{h}_{1d} \cdot \nabla) q)_{\tilde{\Omega}_1} = 0 \\ &\mathbf{u}_1 = \mathbf{u}_2 \quad \text{on } \tilde{\Gamma}_1 \end{aligned} \tag{48}$$

Domain 2

$$\begin{aligned} &(\rho_2 \partial_t \mathbf{u}_2, \mathbf{v})_{\tilde{\Omega}_2} + \langle \rho_2 (\mathbf{u}_2 \cdot \nabla) \mathbf{u}_2, \mathbf{v} \rangle_{\tilde{\Omega}_2} + (\boldsymbol{\sigma}_2, \nabla \mathbf{v})_{\tilde{\Omega}_2} + \tfrac{1}{2} (\mathbf{r}_{2m}, (\mathbf{h}_{2m} \cdot \nabla) \mathbf{v})_{\tilde{\Omega}_2} \\ &\quad + (\bar{\mathbf{t}}, \mathbf{v})_{\Gamma_{2N}} + (\bar{t}_1 \mathbf{g} + \bar{t}_2 \mathbf{s}, \mathbf{v})_{\Gamma_{2M}} + (\tilde{\mathbf{t}}, \mathbf{v})_{\tilde{\Gamma}_2} = \langle \rho_2 \mathbf{f}, \mathbf{v} \rangle_{\tilde{\Omega}_2} \\ &(q, \nabla \cdot \mathbf{u}_2)_{\tilde{\Omega}_2} + \tfrac{1}{2} (r_{2d}, (\mathbf{h}_{2d} \cdot \nabla) q)_{\tilde{\Omega}_2} = 0 \end{aligned} \tag{49}$$

An iterative strategy between the two domains is used to reach a converged global solution in the whole domain Ω. It is expected that this global solution will satisfy both restrictions presented above. For this reason we define the following stop criteria:

$$\begin{aligned} \|\mathbf{u}_1 - \mathbf{u}_2\| &\leqslant \text{tol}_u \quad \text{on } \Omega \\ |\mathbf{n} \cdot (\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_2) \cdot \mathbf{n} - \gamma \kappa| &\leqslant \text{tol}_{\boldsymbol{\sigma}} \quad \text{on } \Gamma \end{aligned} \tag{50}$$

The global velocity solution obtained from Equation (49) is used as the convective velocity for the transport of the level set function, i.e.

$$(\partial_t \psi, \zeta)_\Omega + \langle (\mathbf{u} \cdot \nabla) \psi, \zeta \rangle_\Omega + \tfrac{1}{2} (r_\psi, (\mathbf{h}_\psi \cdot \nabla) \zeta)_\Omega = 0 \tag{51}$$

Unfortunately there is not a theoretical result that can ensure the convergence of this method. However, our experience in applying this method to a wide range of problems involving moving interfaces has showed that the method is stable and robust.

An expected property of the method is its dependency with the mesh size around the interface. This issue is directly related with how good is Γ approximated by $\tilde{\Gamma}_i$, as has been pointed out in property (43). A fine mesh close to the interface significantly reduces the amount of iterations needed to reach a converged global solution, as well as the accuracy of the velocities and the pressure at the interface.

This methodology can be viewed as a combination of domain decomposition and level set techniques. This justifies the name given to the new method: ODDLS.

DISCRETIZATION BY THE FEM

In this section we present the final discrete system of equations associated with problem (48) and (49) using the FEM. Let us consider a uniform partition of the time interval of analysis $[0, T]$, with time step sizeδt. We will denote by a superscript the time step at which the algorithmic solution is computed. We assume that $\mathbf{u}_1^n, p_1^n, \mathbf{u}_2^n$ and p_2^n are known. If $\theta \in [0, 1]$, the trapezoidal rule applied to Equations (48) and (49) consists in finding $\mathbf{u}_1^{n+1}, p_1^{n+1}, \mathbf{u}_2^{n+1}, p_2^{n+1}$ as the solution to the problem

$$\begin{aligned}(\rho_i^n \delta_t \mathbf{u}_i^n, \mathbf{v})_{\tilde{\Omega}_i} + \langle \rho_i^n (\mathbf{u}_i^{n+\theta} \cdot \nabla) \mathbf{u}_i^{n+\theta}, \mathbf{v} \rangle_{\tilde{\Omega}_i} + (\boldsymbol{\sigma}_i^{n+\theta}, \nabla \mathbf{v})_{\tilde{\Omega}_i} + \tfrac{1}{2} (\mathbf{r}_{im}^{n+\theta}, (\mathbf{h}_{im}^{n+\theta} \cdot \nabla) \mathbf{v})_{\tilde{\Omega}_i} \\ + (\bar{\mathbf{t}}, \mathbf{v})_{\Gamma_{iN}} + (\bar{t}_1 \mathbf{g} + \bar{t}_2 \mathbf{s}, \mathbf{v})_{\Gamma_{iM}} + (i-1)(\tilde{\mathbf{t}}, \mathbf{v})_{\tilde{\Gamma}_i} = \langle \rho_i^n \mathbf{f}, \mathbf{v} \rangle_{\tilde{\Omega}_i} \qquad (52) \\ (q, \nabla \cdot \mathbf{u}_i^{n+\theta})_{\tilde{\Omega}_i} + \tfrac{1}{2} (r_{id}^{n+\theta}, (\mathbf{h}_{im}^{n+\theta} \cdot \nabla) q)_{\tilde{\Omega}_i} = 0, \quad i = 1, 2\end{aligned}$$

where $x^{n+\theta} := \theta x^{n+1} + (1-\theta) x^n$ and $\delta_t x^n := 1/\delta t (x^{n+1} - x^n)$.

This problem is non-linear due to the convective term and the evolution of the free surface. Prior to the finite element discretization, we linearize it using a Picard method, which leads to an Oseen problem for each iteration step. Several strategies can be adopted to deal with the two iterative algorithms involved in the ODDLS method: the overlapping domain decomposition iterative scheme and the linearization scheme (Picard). In our case, for each iteration step of the linearization scheme, the domain decomposition scheme is also updated. This simultaneous iterative strategy requires to complete Equation (50) with a stop criterion for the Picard iteration. The same iteration index is used for both schemes.

It is noteworthy that the adopted strategy reduces the computational effort compared with that of other nested iterative schemes. We denote by a superscript n the time step at which the algorithmic solution is computed and by a superscript j the iteration step of the domain decomposition scheme (which for a simultaneous iterative strategy will be the same as for the Picard iteration step). The form of the iterative scheme is as follows:

$$\begin{aligned}(\rho_i^{n,j-1} \delta_t \mathbf{u}_i^{n,j}, \mathbf{v})_{\tilde{\Omega}_i} + \langle \rho_i^n (\mathbf{u}_i^{n+\theta,j-1} \cdot \nabla) \mathbf{u}_i^{n+\theta,j}, \mathbf{v} \rangle_{\tilde{\Omega}_i} + (\boldsymbol{\sigma}_{\boldsymbol{\iota}}^{n+\theta,j}, \nabla \mathbf{v})_{\tilde{\Omega}_i} + \tfrac{1}{2} (\mathbf{r}_{im}^{n+\theta,j}, (\mathbf{h}_{im}^{n+\theta,j} \cdot \nabla) \mathbf{v})_{\tilde{\Omega}_i} \\ + (\bar{\mathbf{t}}, \mathbf{v})_{\Gamma_{iN}} + (\bar{t}_1 \mathbf{g} + \bar{t}_2 \mathbf{s}, \mathbf{v})_{\Gamma_{iM}} + (i-1)(\tilde{\mathbf{t}}^j, \mathbf{v})_{\tilde{\Gamma}_i} = \langle \rho_i^{n,j-1} \mathbf{f}, \mathbf{v} \rangle_{\tilde{\Omega}_i} \qquad (53) \\ (q, \nabla \cdot \mathbf{u}_i^{n+\theta,j})_{\tilde{\Omega}_i} + \tfrac{1}{2} (r_{id}^{n+\theta,j}, (\mathbf{h}_{im}^{n+\theta,j} \cdot \nabla) q)_{\tilde{\Omega}_i} = 0, \quad i = 1, 2\end{aligned}$$

The Galerkin approximation of Equation (53) is straightforward and stable, thanks to the FIC-stabilized formulation [13–18]. Equal polynomial order can be used for the approximation of the velocities and the pressure. Based on the finite element partition introduced in Equation (38), let $V_h \subset V$ and $Q_h \subset Q$ be two finite dimensional conforming finite element spaces, h being the maximum of the elements' partition diameters.

Let $\tilde{V}^0_{h,i} := \{\mathbf{v} \in V_h | \mathbf{v}|_{\partial\tilde{\Omega}_i} = \mathbf{0}\}$, $Q^0_{h,i} := \{q \in Q_h | q|_{\partial\tilde{\Omega}_i} = 0\}$ be the finite element spaces for the test functions. The discretized form of Equations (55) is

$$
\begin{aligned}
&(\rho^{n,j-1}_{h,i}\delta_t \mathbf{u}^{n,j}_{h,i}, \mathbf{v}_h)_{\tilde{\Omega}_i} + \langle \rho^n_{h,i}(\mathbf{u}^{n+\theta,j-1}_{h,i} \cdot \nabla)\mathbf{u}^{n+\theta,j}_{h,i}, \mathbf{v}_h\rangle_{\tilde{\Omega}_i} + (\boldsymbol{\sigma}^{n+\theta,j}_{h,i}, \nabla \mathbf{v}_h)_{\tilde{\Omega}_i} \\
&\quad + \tfrac{1}{2}(\mathbf{r}^{n+\theta,j}_{h,im}, (\mathbf{h}^{n+\theta,j}_{h,im} \cdot \nabla)\mathbf{v}_h)_{\tilde{\Omega}_i} + (\bar{\mathbf{t}}, \mathbf{v}_h)_{\Gamma_{iN}} + (\bar{t}_1 \mathbf{g} + \bar{t}_2 \mathbf{s}, \mathbf{v}_h)_{\Gamma_{iM}} \\
&\quad + (i-1)(\tilde{\mathbf{t}}^j, \mathbf{v}_h)_{\tilde{\Gamma}_i} = \langle \rho^{n,j-1}_{h,i} \mathbf{f}, \mathbf{v}_h\rangle_{\tilde{\Omega}_i} \\
&(q_h, \nabla \cdot \mathbf{u}^{n+\theta,j}_{h,i})_{\tilde{\Omega}_i} + \tfrac{1}{2}(r^{n+\theta,j}_{h,id}, (\mathbf{h}^{n+\theta,j}_{h,im} \cdot \nabla) q_h)_{\tilde{\Omega}_i} = 0, \\
&\quad i = 1, 2 \;\; \forall (\mathbf{v}_h, q_h) \in \mathbf{V}_h \times Q_h
\end{aligned}
\tag{54}
$$

Remark
Updating the level set equation is done once the convergence of the Navier–Stokes equations is obtained. Therefore, the complete iteration loop is as follows:

Step 1: Solve Equations (54) in $\tilde{\Omega}_1$.

Step 2: Solve Equations (54) in $\tilde{\Omega}_2$.

Step 3: Check convergence on pressure and velocity fields. If it is satisfied, go to Step 4, otherwise go to Step 1.

Step 4: Solve Equation (51).

Step 5: Check convergence on level set equation. If it is satisfied, advance to the next time step, otherwise go to Step 1.

ALE FORMULATION

It is of interest in many applications to consider the movement of some parts of the analysis domain. In the mobile parts of the domain it is more convenient to use a Lagrangian formulation of the equations and update the spatial discretization every time step. Conversely, in the fixed areas of the analysis domain, it is more efficient to use the standard Eulerian formulation.

The ALE formulation of the residuals in (35) can be expressed as follows:

$$
\begin{aligned}
\mathbf{r}_m &= \partial_t(\rho \mathbf{u}) + \nabla \cdot (\rho(\mathbf{u} - \mathbf{u}^m) \otimes \mathbf{u}) + \nabla p - \nabla \cdot \boldsymbol{\sigma} - \rho \mathbf{f} \\
r_d &= \nabla \cdot \mathbf{u} \\
r_\psi &= \partial_t \psi + ((\mathbf{u} - \mathbf{u}^m) \cdot \nabla)\psi
\end{aligned}
\tag{55}
$$

where $\mathbf{u}^m$ is the relative velocity between the local axes fixed to the fluid particle and the global reference axes of the problem [22, 23].

ADAPTATION OF THE ODDLS METHOD FOR MONOPHASE FLOWS

It is usual in many practical applications to have only one fluid of interest. These applications involve biphase flows with density and viscosity ratios about 1000 and 100, respectively. It is important for these cases to adapt the ODDLS technique to solve monophase problems in order to reduce the computational cost for capturing the free surface with the necessary accuracy, while maintaining the advantages of the proposed method. In these cases, the computational domain is reduced to the nodes in the fluid of interest in addition to those in the second fluid being connected to the interface. The later nodes are used to impose the pressure and velocity boundary conditions on the interface.

The proposed monophase adaptation of the ODDLS method has been used in some of the application examples shown in the following section.

APPLICATION EXAMPLES

Different validation studies of numerical results obtained with the methodology described here are presented. In order to perform these analyses, the ODDLS methodology has been integrated into the commercial pre- and postprocessing system GiD [24] and the fluid-dynamics analysis package Tdyn [25].

The geometry and results of the analyses presented in this section as well as a free limited version of the software Tdyn can be downloaded from ftp://ftp2.compassis.com/papermodels.

Wave generator

The first example shows a comparison of the results obtained with a numerical wave generator with those predicted by linear wave theory.

The geometry used for the analysis consists of a 2D wave tank of 36 m long filled up to a depth of 1.25 m (see Figure 2). The waves are generated by defining an oscillating velocity boundary condition on the left lateral wall of the channel. The velocity is obtained from the equivalent movement of a wall given by $x = d \cdot \sin(\omega t)$, where ω is the wave generator frequency and d is the amplitude of the movement (stroke). The waves are dumped at the right-hand side of the tank by disposing a beach.

The mesh chosen has 34 424 three-noded linear triangles with a homogeneous size distribution on the whole domain.

For this problem linear wave theory states that the relationship between wave number, channel depth and wave-maker frequency is as follows:

$$\omega^2 = gk \tanh(kh) \tag{56}$$

where k is the wave number, h is the depth of the channel and g is the gravity acceleration.

Several analyses were carried out with a wave generator period ranging from $T = 1.8$ to 3.0 s.

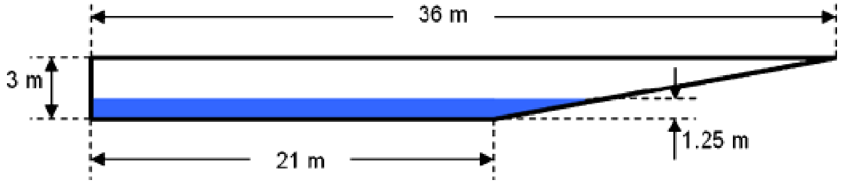

Figure 2. Geometry of the wave generator problem.

Figure 3 shows a sequence of snapshots of the wave maps obtained for $T = 2.0\,\mathrm{s}$ with a stroke of 0.3.

Figure 4 shows the distribution of the dynamic pressure for this problem. The wave length computed is approximately 5.3 m, whereas the linear wave theory predicts a value of 5.5 m.

Figure 5 shows a sequence of four snapshots of the resulting wave maps for a wave-maker period of $T = 2.1\,\mathrm{s}$ and a stroke of 0.4 m. Linear wave theory predicts a wavelength of 5.9 m, which closely matches the simulation value (5.7 m).

Figure 6 shows a comparison between the different wavelength values ($\lambda = 1/k$) obtained in the numerical analyses and those predicted by the linear wave theory. The wave generator period analysed ranges from $T = 1.8$ to 3.0 s and all the cases were run with a stroke of 0.3 m. The ODDLS results agree quite well with those predicted by the theory. The small differences found are probably due to the non-linear effects and the dissipative influence of the viscosity.

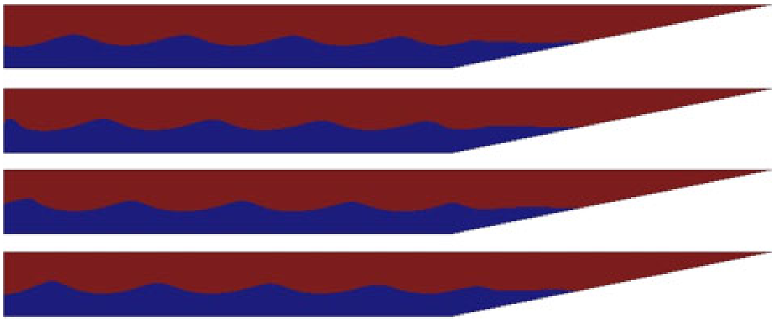

Figure 3. Wave map evolution for $T = 2.0\,\mathrm{s}$.

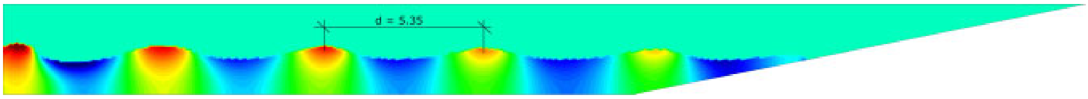

Figure 4. Dynamic pressure distribution for $T = 2.0\,\mathrm{s}$.

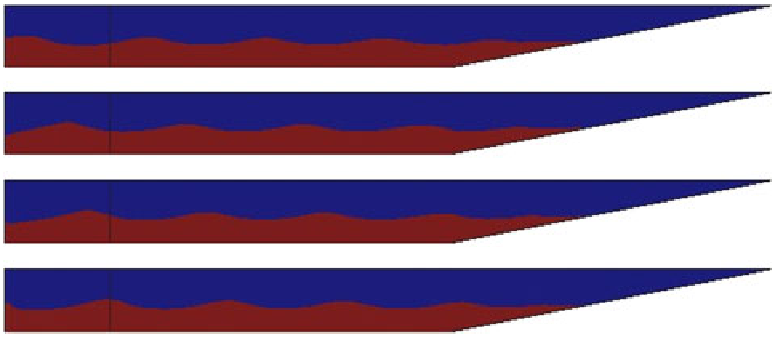

Figure 5. Wave map evolution for $T = 2.1\,\mathrm{s}$.

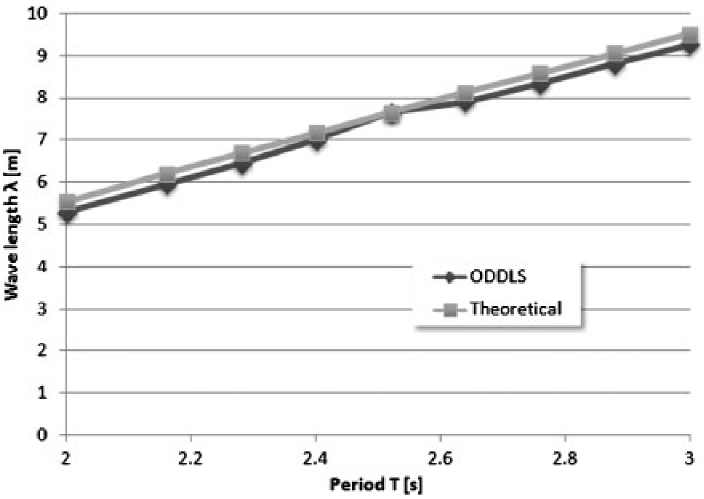

Figure 6. Comparison between numerical and (ODDLS) theoretical values of the wavelength.

Figure 7 shows the dynamic pressure maps obtained after 21 s of physical time for different wave-maker periods and 0.3 m stroke.

Additionally, the convergence behaviour of the method under mesh refinement has been studied. The case $T = 2.6$ s was run with three different meshes: 35 000 linear triangles, 140 000 linear triangles ($\frac{1}{2}$ of the previous global mesh size) and 560 000 linear triangles ($\frac{1}{4}$ of the previous global mesh size). The results obtained in these simulations are shown in Figure 8. This figure shows a snapshot of velocity contours at $t = 14$ s. The wave pattern in the three cases is practically identical.

Finally, an additional case has been analysed in order to check the influence of the reinitialization step in the results. In this case, an obstacle of 1×1 m has been arranged at 10 m from the inlet. This obstacle causes the generated wave ($T = 2.6$ s in this case) to break as shown in Figure 9. In this figure, the results above correspond to the case where the reinitialization step is performed every iteration, whereas the results below correspond to the case where the reinitialization step is performed every five iterations. These results show a minor sensitivity with respect to the number of reinitializations. However, more reinitializations tend to slightly smoothen the interface and to remove sharp features from the interface geometry.

Flip bucket

This example shows an application of the ODDLS methodology to the analysis of a flip bucket. Flip buckets are commonly used to discharge flow away from a hydraulic structure into a plunge pool to dissipate energy. The ODDLS method results are compared with those obtained in the hydraulic experiments presented in [5]. The experiments were conducted in a 499-mm-wide and 700-mm-deep rectangular channel with a total length of 7 m. A scheme of the experimental set-up is shown in Figure 10. The flip bucket consisted of a 1-m-long approach channel with a bucket of radius R and a deflection angle β. The solution to the same problem using the PFEM based on a purely Lagrangian formulation is reported in [20].

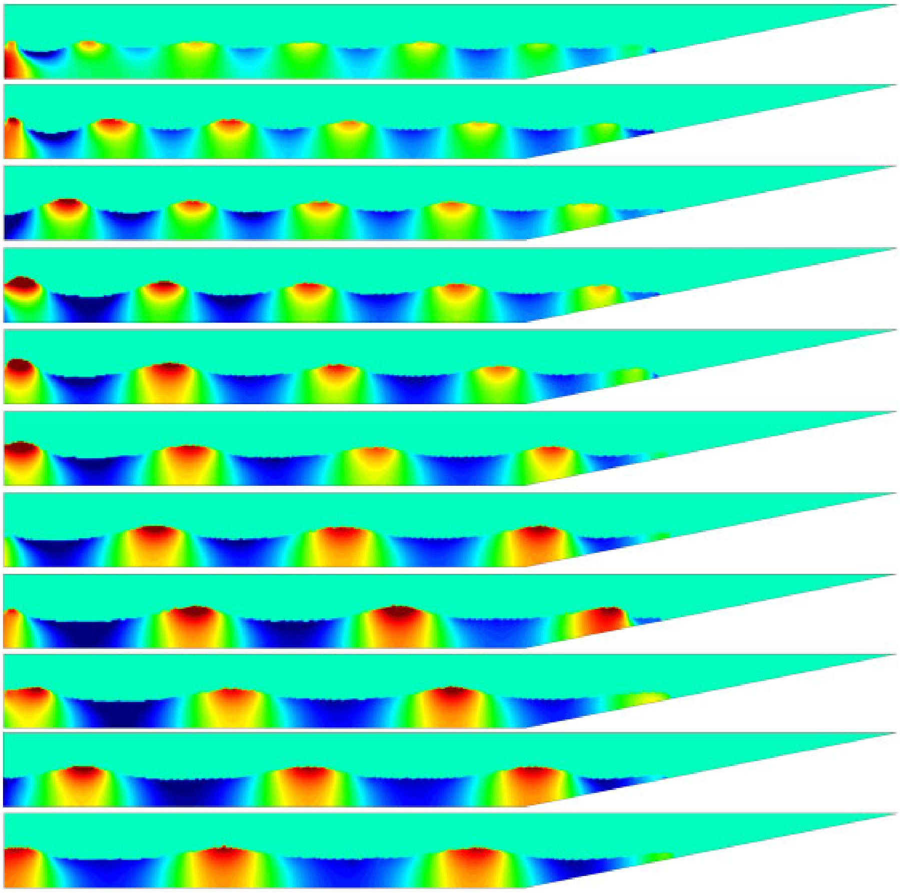

Figure 7. Dynamic pressure maps obtained after 21 s of physical time for different simulations (from bottom to top $T = 3.0, 2.88, 2.76, 2.64, 2.52, 2.4, 2.28, 2.16, 2.04, 1.92, 1.80$ s).

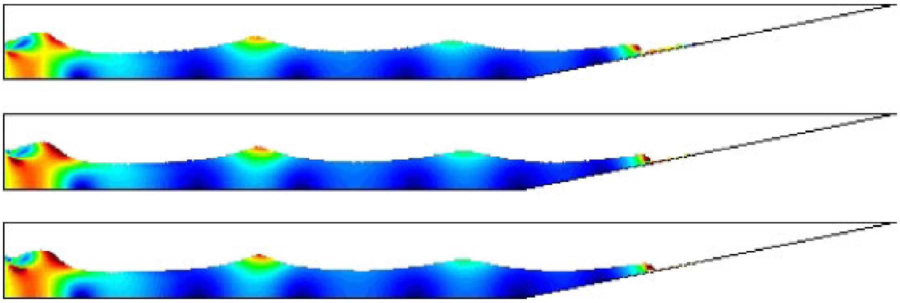

Figure 8. Comparison of results obtained in the case $T = 2.6$ s (snapshot of velocity contours in the water at $t = 14$ s) with three different meshes (from top to bottom 35 000 linear triangles, 140 000 linear triangles and 560 000 linear triangles).

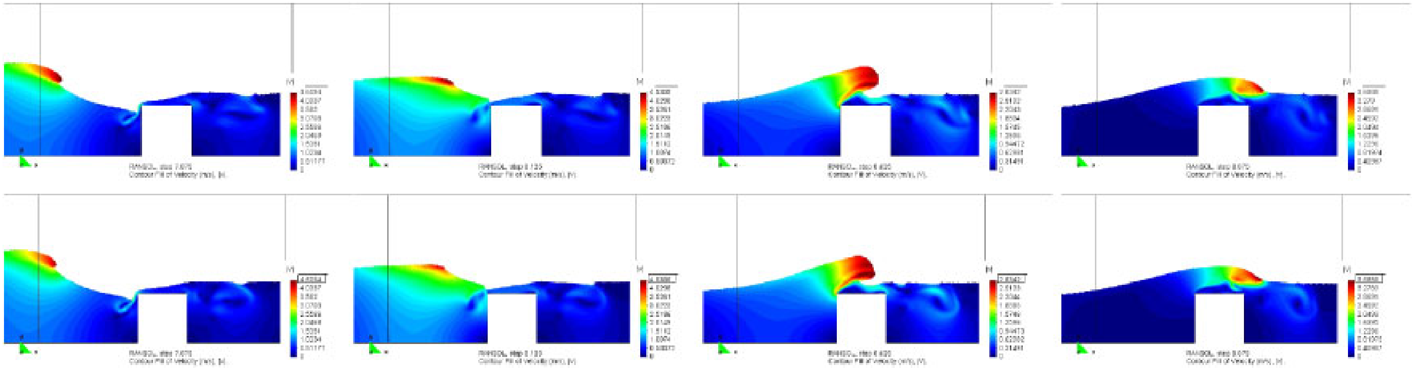

Figure 9. Comparison of results (velocity contours in the water) obtained in the case $T = 2.6\,\mathrm{s}$ with an obstacle with the reinitialization step performed every iteration (above) or every five iterations (below).

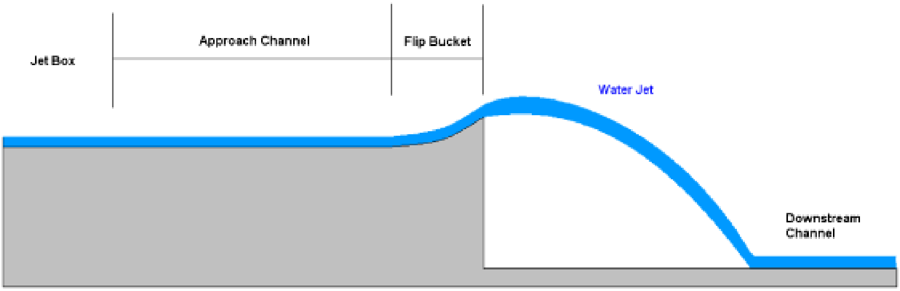

Figure 10. Sketch of the experimental set-up (lateral view).

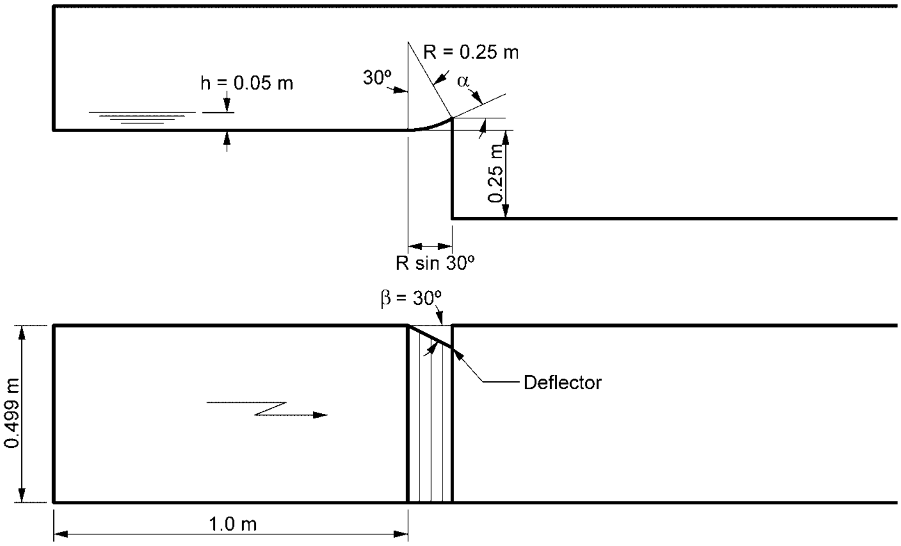

Figure 11. Dimensions of the experimental facilities (lateral view, above; top view, below).

Figure 11 summarizes the main dimensions of the experimental facility. The geometry used for the numerical simulations reproduces the experimental set-up. Only half of the channel geometry was defined, due to the symmetry of the problem.

The mesh used is shown in Figure 12. It consists of 1.5 million four-noded linear tetrahedra and around 275 000 nodes. The element size ranges from 1 to 5 cm. Note that the mesh has been refined in a band around the area where water is expected to flow.

The analyses have been carried out with and without the deflector, for two different arc radii of the flip bucket and four different Froude numbers $Fn = 3.0, 4.0, 5.0$ and 7.0, with the Froude number defined as

$$Fn = \frac{\bar{u}_{\text{in}}}{\sqrt{gh}} \tag{57}$$

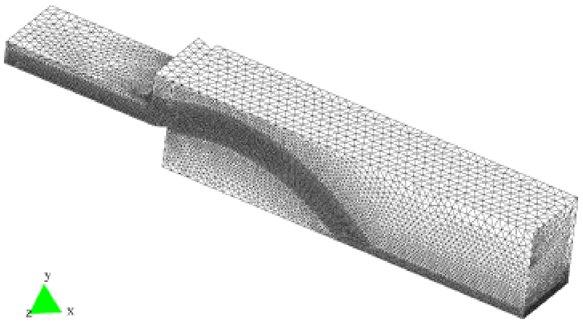

Figure 12. Mesh used in the different analyses.

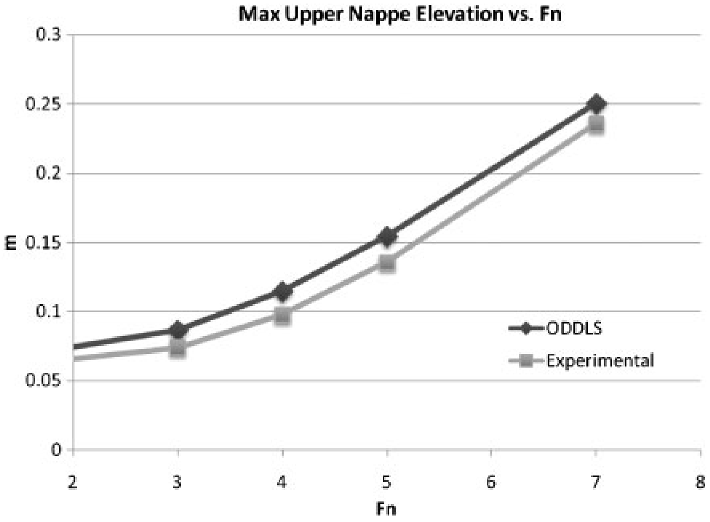

Figure 13. Maximum set elevation *versus* Froude number.

where $\bar{u}_{\text{in}}$ is the mean flow velocity in the approaching channel and h is the depth of fluid. All the simulations have been carried out with a fluid density $\rho = 1000\,\text{kg/m}^3$ and a viscosity $\mu = 0.001\,\text{kg/ms}$.

Figure 13 compares the maximum experimental set elevation with the values obtained in the simulation for each of the four Froude numbers analysed. The ODDLS results closely match the experimental data in all cases.

Figure 14 shows a comparison between the numerical and experimental dynamic pressure head distributions at the bottom of the entry channel for $R = 0.25\,\text{m}$. The non-dimensional dynamic pressure head parameter is defined as

$$H_p = \frac{p_p - p_0}{p_{\max} - p_0} \tag{58}$$

where p_p is the total pressure head at a point in the entry channel or the bucket, $p_{\max}$ is the maximum value of p and p_0 is the total pressure head at the entry point. The normalized streamwise

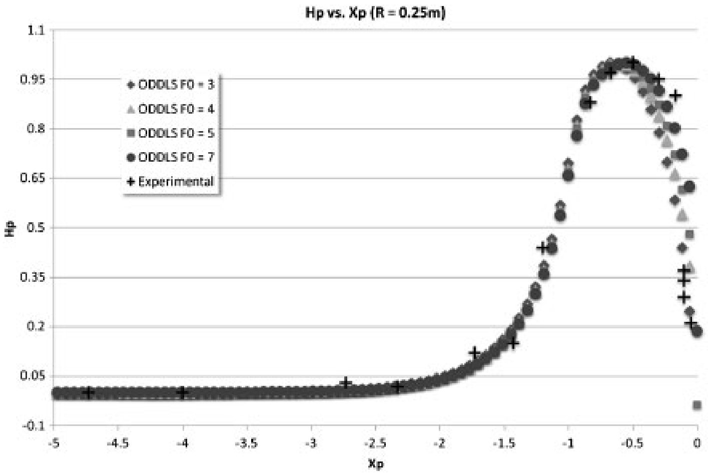

Figure 14. Comparison between the numerical and experimental bottom pressure distributions [5] for $R = 0.25\,\mathrm{m}$.

coordinate used in Figure 14 is defined as $X_P = x/(R\sin\beta)$, where x is zero at the takeoff point; and $R\sin\beta$ is the streamwise bucket length.

Once again the numerical results agree quite well with the experimental data available.

Because of the deflector, some of the water moves upwards, forming the so-called *shock wave*. As observed in the experiment, the shock wave forms slightly upstream of the deflector, forces its way upwards and expands transversely after the minimum jet width point before touching down in the downstream channel. The maximum height of the shock wave observed in the experiment is about twice that of the unperturbed flow [5].

Figure 15 shows a comparison between the numerical (ODDLS) and experimental shock wave profiles for $F_0 = 4$. The non-dimensional parameters X_S, Y_S are defined as

$$X_S = \frac{x - x_{s\,\max}}{hFn}, \quad Y_S = \frac{y_s - y_{s\,\max}}{h\sin^{3/8}\eta} \tag{59}$$

where y_s is the water jet elevation relative to the takeoff point, $y_{s\,\max}$ is the maximum of y_s, η is the deflection angle (30° in our case), x is the longitudinal distance from the takeoff point, $x_{s\,\max}$ is the longitudinal distance of the maximum shock wave elevation from the takeoff point and h_0 is the water depth at the entry channel, 0.05 m.

Finally, Figure 16 shows a sequence of snapshots of the jet surface for the case with a deflector, $Fn = 4$ and $R = 0.25\,\mathrm{m}$.

2D Dam-break

The 2D dam-break case has been analysed experimentally by Zhou *et al.* [26] and numerically with the boundary element method by Greco [27]. In this example the numerical pressure obtained by the later is compared with the ODDLS results. The problem set-up is shown in Figure 17.

In order to compare the BEM results with the experimental results, Greco had to synchronize the time series due to lack of information about the initial dam-break time in the experiment. Greco

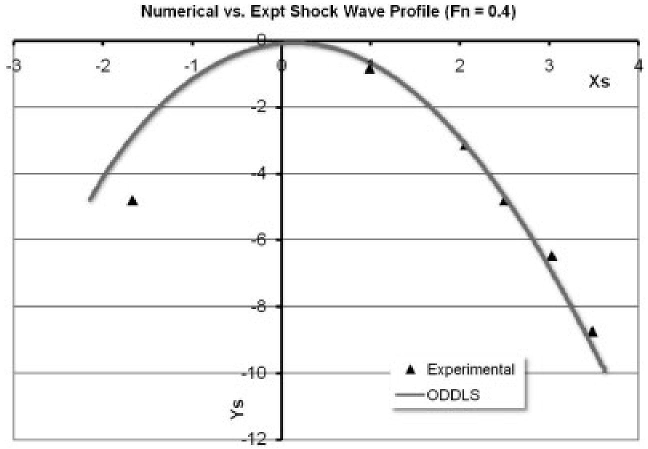

Figure 15. Numerical *versus* experimental [5] shock wave profile for $Fn = 4$.

also points out some uncertainties connected to the complexity of the experiment and the difficulty in achieving repeatability. In what follows we compare the ODDLS results with the BEM results reported in [27].

The ODDLS has 127 000 triangular elements in an unstructured mesh. On the walls the slipping boundary condition is imposed, neglecting boundary layer effects, and the fluid is fresh water at 25°C. The effect of air is neglected. The comparison of non-dimensional pressures at the two gauge positions are shown in Figure 18 (left). The results obtained fit well with those of Greco, although a small difference in the time of impact on the wall is perceived.

The BEM analysis had to be stopped when the water plunged down onto itself after the run up on the wall. The ODDLS method is able to handle this situation. The peak in Figure 16 (right) relates to this time instant and it agrees well with the experimental data reported in [26].

Figure 19 compares the wave profile at different time instants. Once again the ODDLS solution is a bit slower than that of Greco. This effect is probably due to the viscous effects that are neglected in the BEM analysis.

2D Green water problem

The example shows an application of the ODDSL method to the collapse of a water column with an obstacle. The geometry of the tank is shown in Figure 20. This example can be seen as a simple model of green water flow on the deck of a ship or offshore structure. The obstacle placed on the bottom of the tank represents an element of the ship superstructure.

Experimental results for this problem have been presented in [28]. The water column is supported by a vertical wall, which is removed rapidly (approximately in 0.05 s) for the beginning of collapse.

The mesh used to discretize the analysis domain has 35 000 linear triangles. The fluid is incompressible with a density of $1000\,\text{kg/m}^3$ and a viscosity of 0.001 kg/ms. The slipping boundary condition is assumed on the walls of the tank. Zero velocity has been imposed on the obstacle.

Figure 21 shows different pictures of the experiment results. At time 0.1 s, the right surface of the water column is disturbed by the pulling up of the vertical wall in the experiment, although it is soon smoothed. At time 0.2 s, the leading edges of the water column reach $3l$ from the left

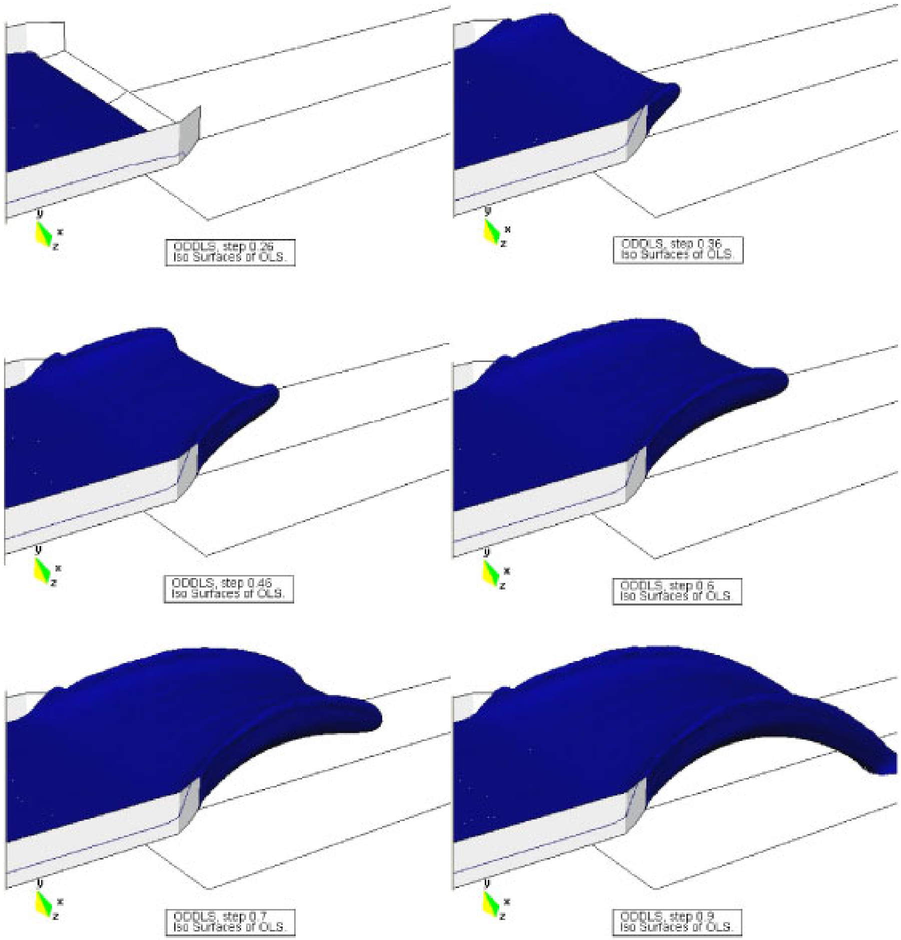

Figure 16. Sequence of snapshots of the jet surface for the case with a deflector, $Fn = 4$ and $R = 0.25\,\text{m}$.

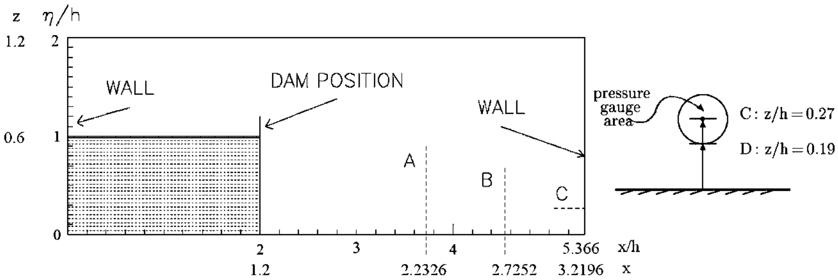

Figure 17. Sketch of the 2D dam-break problem with $h = 0.6\,\text{m}$. The pressure gauge at position $x = 3.2196\,\text{m}$ is shown in [26].

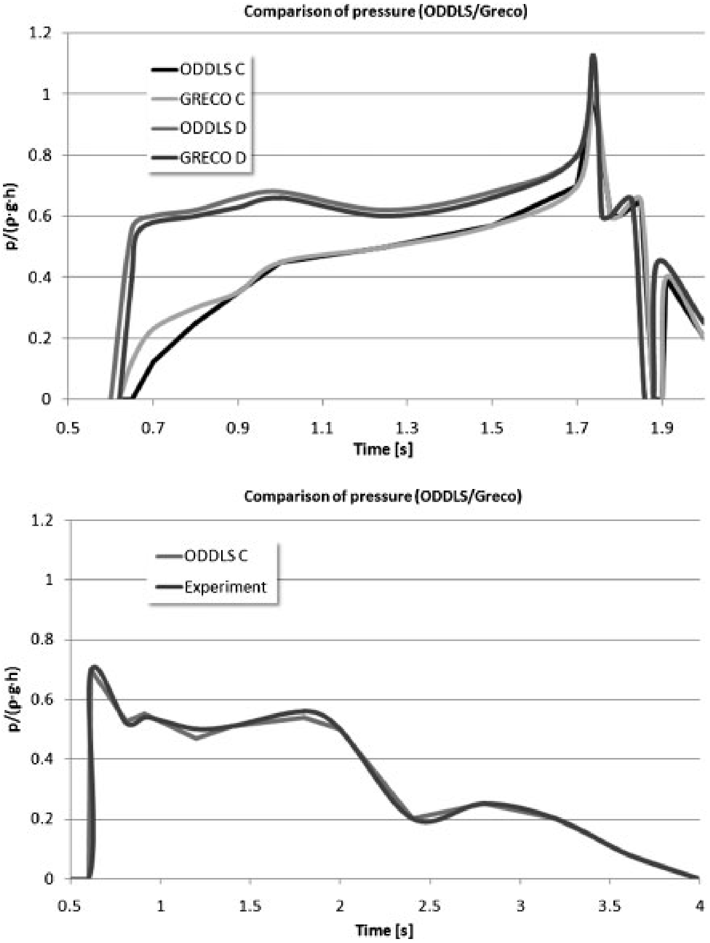

Figure 18. Comparison of pressures at the two gauge positions.

wall. At time 0.3 s the collapsed water is very close to the right and it crashes immediately. At time 0.4 s, splashed water goes up along the right wall with fragmentation.

As in the wave breaking case presented above, this case has been additionally run performing the reinitialization procedure every five time steps. No noticeable changes in the solution have been detected. A comparison of the results at time 0.4 s of the two cases are shown.

Figure 21 shows good agreement between the experimental and numerical results obtained for the presented ODDLS method.

3D Green water problem

Experiments have been performed at the Maritime Research Institute Netherlands (MARIN) in order to reproduce a simple model of green water flow on the deck of a ship or an offshore structure [29]. Green water can cause serious damage to deck structures, deck house, cargo and personnel.

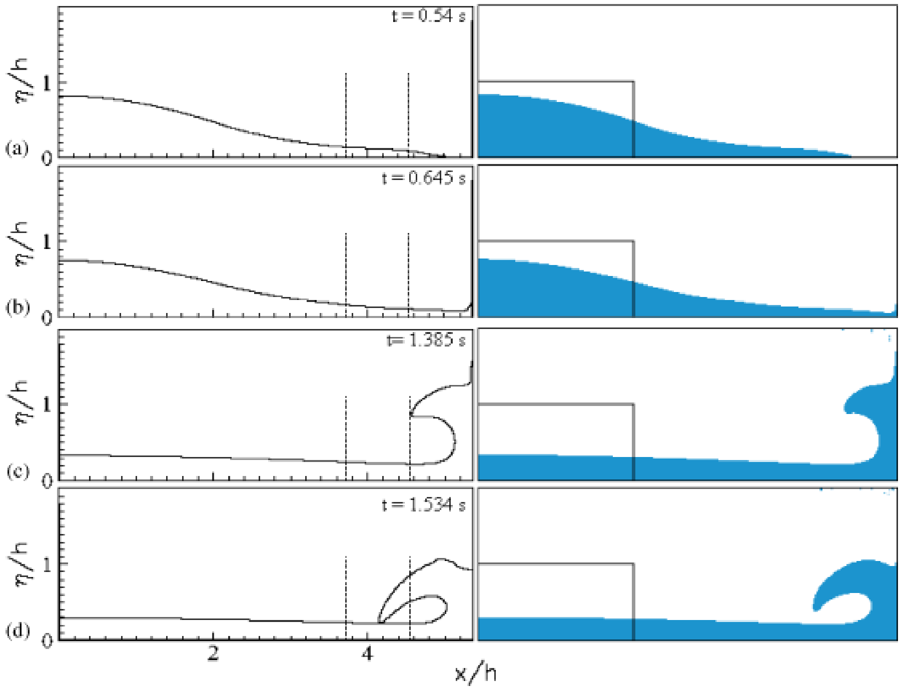

Figure 19. Comparison of wave profiles. Profiles from Greco [27] and ODDLS (right).

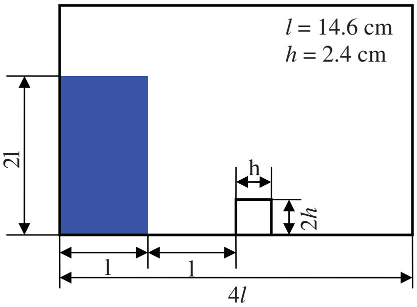

Figure 20. Geometry of the 2D green water problem.

In the test, an object, resembling an element of the superstructure, is placed on the deck. The pressures that are exerted by the plunging dam are measured as a function of time. The model is described in Figure 22 and consists of a tank with an open roof of dimensions $3.22 \times 1 \times 1$ m. The dam has a height of 0.55 m and is closed behind a door. The door is opened almost instantaneously by releasing a weight. The locations of the pressure gauges are shown in Figure 23. This problem has previously been used to validate a volume-of-fluid method [30] and the particle finite element method [20].

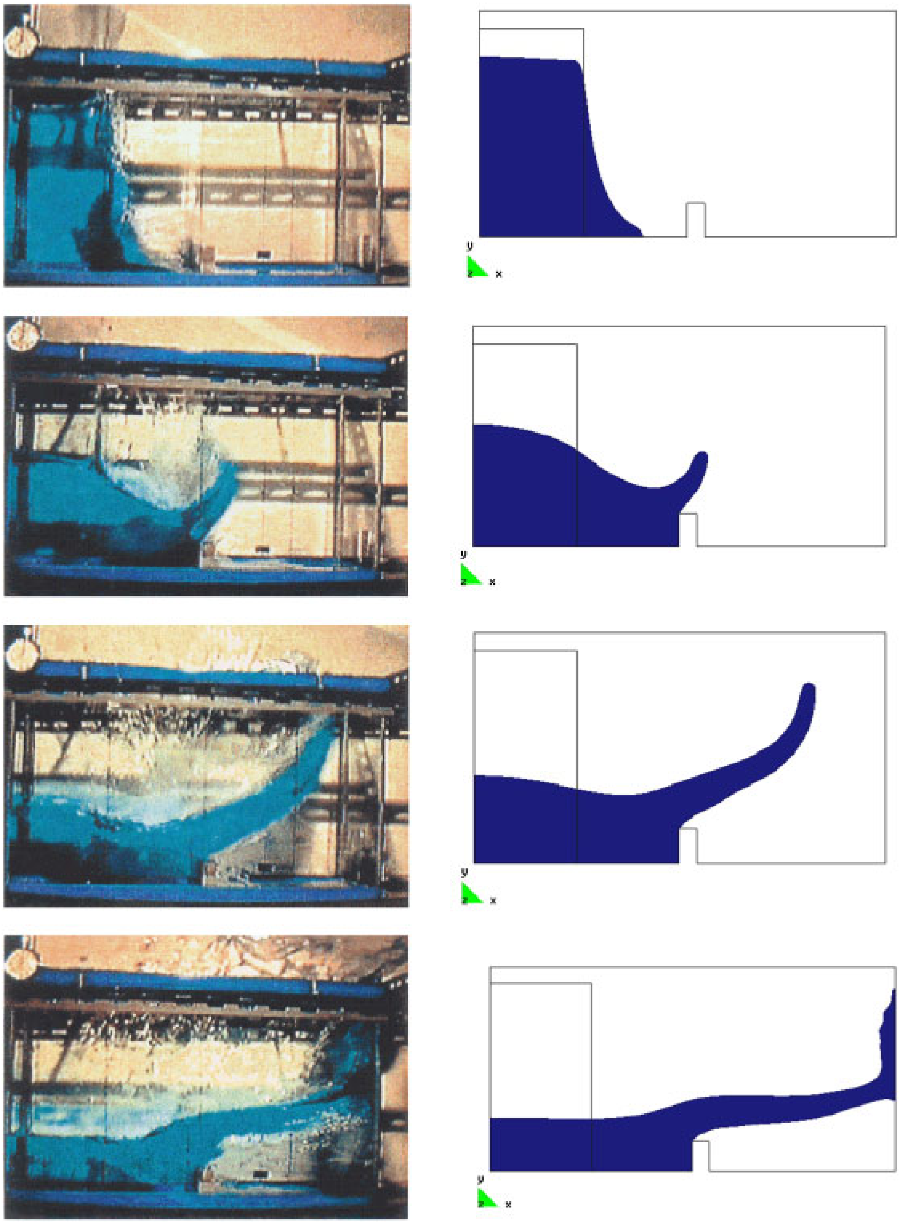

Figure 21. Experimental (left) and numerical results (right).

The fluid is incompressible with a density of 1000 kg/m^3 and a viscosity of 0.001 kg/ms. The slipping boundary condition is imposed on the walls of the tank. The model consists of 1.16 million tetrahedra in an unstructured mesh. Figure 24 shows a comparison between the zero level set function, the free surface and the experimental water front. The pressures at four of the gauges

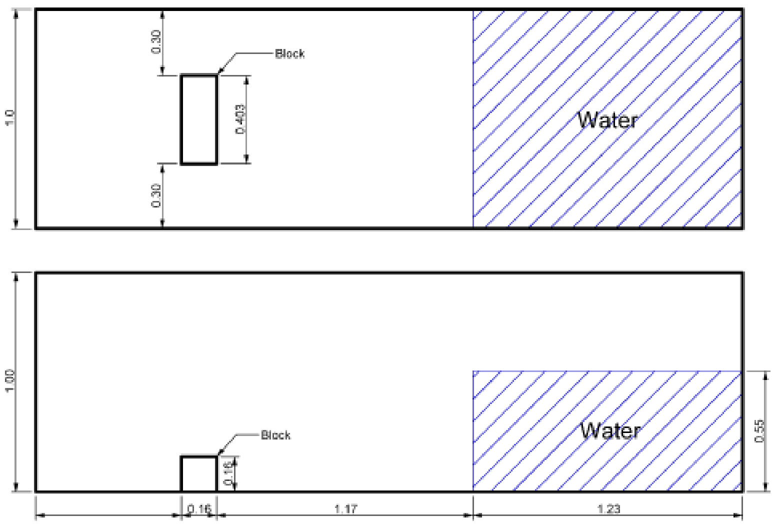

Figure 22. Dimensions of dam-break tank [29].

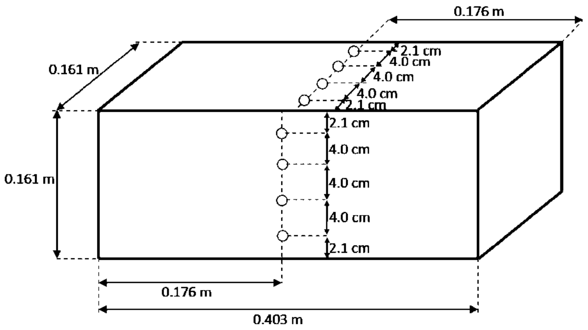

Figure 23. Positions of pressure gauges [29].

are compared with the ODDLS results in Figure 25. P1 and P2 are positioned close to the initial impact point, whereas P6 and P7 are positioned at the top of the object. There is an excellent agreement between the experimental and ODDLS results in all cases, although the pressure peaks obtained in P1 and P2 are slightly higher than the experimental values.

Finally, Figure 26 shows the evolution of the free surface at several times.

CONCLUSIONS

A new methodology for the analysis of free surface flows termed ODDLS has been presented. The method is based on the domain decomposition technique combined with the level set technique and a stabilized FEM. The ODDLS approximation allows a sharper resolution of contact discontinuities between different fluid phases compared with more standard methods, such us volume of fluid,

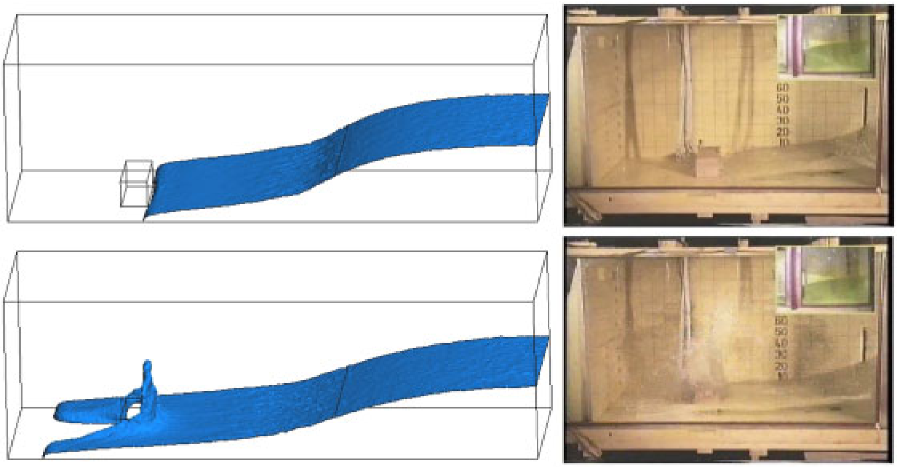

Figure 24. Zero level set surface at $t = 0.425\,\mathrm{s}$ (upper) and $t = 0.575\,\mathrm{s}$ (lower).

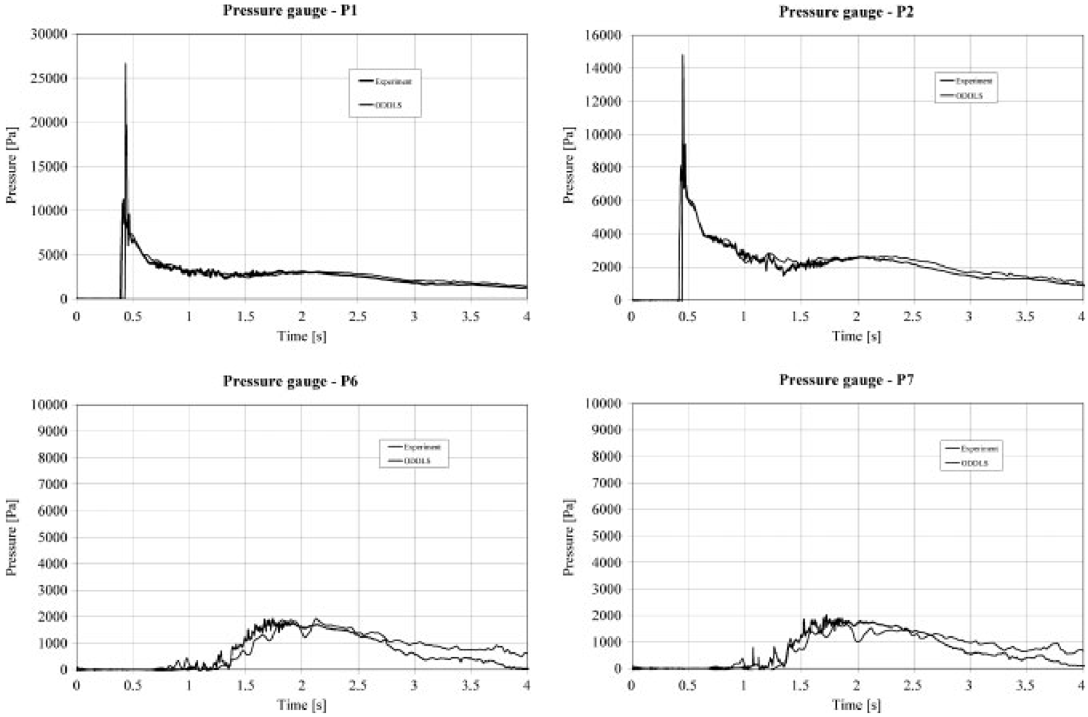

Figure 25. Comparison of experimental [29] and ODDLS pressures at positions P1, P2, P6 and P7.

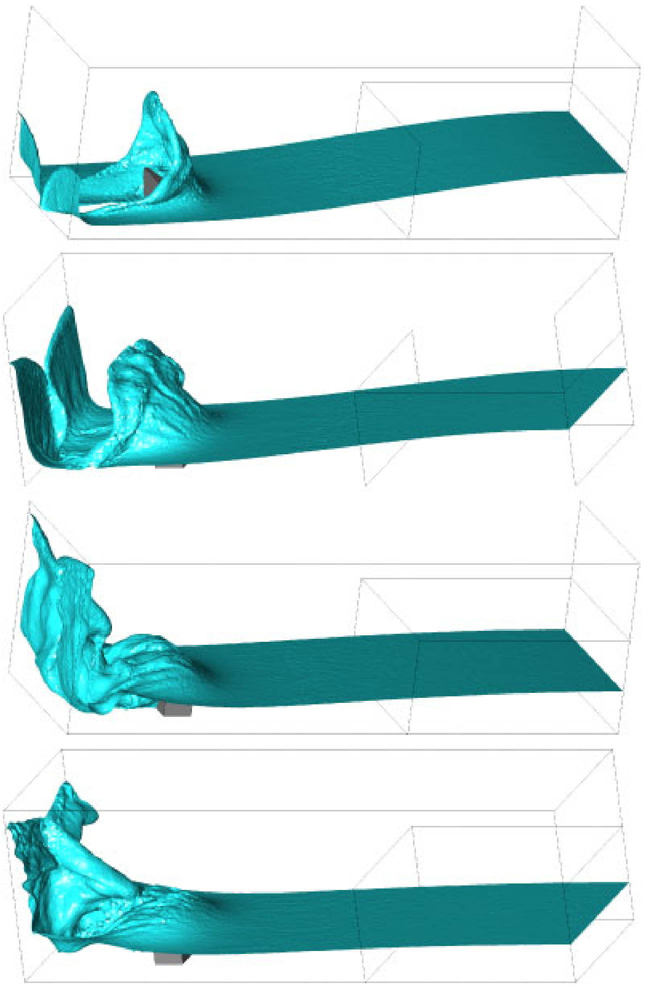

Figure 26. Zero level set surface at $t = 0.8, 1.0, 2.0$ and $2.3\,\mathrm{s}$.

where the variation of physical properties of the phases in contact must be smoothed. This fact increases the accuracy of the free surface capturing (level set equation) as well as the solution to the governing equations in the interface between two fluids. The higher accuracy in the solution to the interface allows the use of larger elements on the free surface. The method can be easily simplified by solving just one of the two fluids, which increases the efficiency in cases where the effect of one of the fluids can be neglected.

The ODDLS methodology has also been integrated with an ALE algorithm for the treatment of moving meshes.

The ODDLS technique has been applied for the analysis of several free surface flows problems. The good qualitative and quantitative results obtained in comparison with experimental data and other numerical techniques show the capability of the ODDLS methodology for solving free surface flow problems of practical interest.

ACKNOWLEDGEMENTS

The authors are grateful to Mr Simen Fodstad Stølen for helping in the computation of several of the examples presented here.

This work was partially supported by the projects TRA2005-07536 07536/TMAR, VEM2004-08641-C03-03 and SEDUREC (Consolider INGENIO 2010 Programme) of the Ministerio de Educación y Ciencia, Spain.

REFERENCES

1. Monaghan JJ. Simulating free surface flows with SPH. *Journal of Computational Physics* 1994; **110**(2):399–406.
2. *Proceedings of the 1st Spheric Workshop*, Rome, May 2006. http://w3.uniroma1.it/cmar/SPHERIC/SPHERICWorkshop.htm.
3. Oñate E, Idelsohn SR, Del Pin F, Aubry R. The particle finite element method. An overview. *International Journal for Computational Methods* 2004; **1**(2):267–307.
4. Idelsohn SR, Oñate E, Del Pin F, Calvo N. Fluid–structure interaction using the particle finite element method. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:2100–2113.
5. Juon R, Hager WH. Flip bucket with and without deflectors. *Journal of Hydraulic Engineering* 2000; **126**: 837–845.
6. Sethian JA, Smereka P. Level set methods for fluid interfaces. *Annual Review of Fluid Mechanics* 2003; **35**:341–372.
7. Sethian JA. Evolution, implementation, and application of level set and fast marching methods for advancing fronts. *Journal of Computational Physics* 2001; **169**:503–555.
8. Osher SJ, Sethian JA. Front propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics* 1988; **79**:12–49.
9. Osher SJ, Fedkiw RP. Level set methods: an overview and some recent results. *Journal of Computational Physics* 2001; **169**:463–502.
10. Fedkiw RP, Aslam T, Merriman B, Osher SJ. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics* 1999; **154**:393–427.
11. Tezduyar TE, Aliabadi S, Behr M. Enhanced-discretization interface-capturing technique (EDICT) for computation of insteady flow with interface. *Computer Methods in Applied Mechanics and Engineering* 1998; **155**:235–249.
12. Oñate E. Derivation of stabilized equations for advective–diffusive transport and fluid flow problems. *Computer Methods in Applied Mechanics and Engineering* 1998; **151**:233–267.
13. Oñate E. A stabilized finite element method for incompressible viscous flows using a finite increment calculus formulation. *Computer Methods in Applied Mechanics and Engineering* 2000; **182**(1–2):355–370.
14. García Espinosa J, Oñate E. An unstructured finite element solver for ship hydrodynamics. *Journal of Applied Mechanics* 2003; **70**:18–26.
15. Oñate E, García J, Idelsohn S. Ship hydrodynamics. In *Encyclopedia of Computational Mechanics*, Stein E, De Borst R, Hughes TJR (eds). Wiley: New York, 2004.
16. García Espinosa J, Oñate E, Bloch Helmers J. Advances in the finite element formulation for naval hydrodynamics problems. *International Conference on Computational Methods in Marine Engineering* (*Marine 2005*), Oslo, Norway, June 2005.
17. Oñate E, Valls A, García J. FIC/FEM formulation with matrix stabilizing terms for incompressible flows at low and high Reynold's numbers. *Computational Mechanics* 2006; **38**(4–5):440–455.
18. Oñate E, Valls A, García J. Computation of turbulent flows using a finite calculus-finite element formulation. *International Journal for Numerical Methods in Fluids* 2007; **54**(6–8):609–637.
19. Soto O, Lohner R, Cebral J. An implicit monolithic time accurate finite element scheme for incompressible flow problems. *15th AIAA Computational Fluid Dynamics Conference*, Anaheim, U.S.A., 2001.
20. Larese A, Rossi R, Oñate E, Idelsohn SR. Validation of the particle finite element method (PFEM) for simulation of free surface flows. *Engineering Computations* 2008; **25**(4). Paper No. 1246.
21. Quarteroni A, Valli A. *Domain Decomposition Methods for Partial Differential Equations*. Numerical Mathematics and Scientific Computation. Oxford University Press: Oxford, 1999.
22. Hirt CW, Amsden AA, Cook JL. An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *Journal of Computational Physics* 1974; **14**:227–253.

23. Tezduyar TE. Finite element method for flow problems with moving boundaries and interfaces. *Archives of Computational Methods in Engineering* 2001; **8**:83–130.
24. GiD. *The Personal Pre/postprocessor User Manual*, 2007. Available from: http://www.gidhome.com.
25. Tdyn. *Theoretical Background and Reference Manual*, 2007. Available from: http://www.compassis.com.
26. Zhou ZQ, Kat JQD, Buchner B. A nonlinear 3-d approach to simulate green water dynamics on deck. In *Proceedings of 7th International Conference on Numerical Ship Hydrodynamics*, Piquet J (ed.), Nantes, France, 1999; 5.1–1.15.
27. Greco M. A two-dimensional study of green-water loading. *Ph.D. Thesis*, Department of Marine Technology, NTNU, 2001.
28. Koshizuka S, Tamako H, Oka Y. A particle method for incompressible viscous flow with fluid fragmentation. *Journal of Computational Fluid Dynamics* 1995; **4**(1).
29. Issa R, Violeau D. *Test-case 2*, *3D Dambreaking*, *Release 1.1 ERCOFTAC*. SPH European Research Interest Community SIG, Electricite De France, Laboratoire National Hydaulique et Environnement, 2006.
30. Kleefsmann KMT, Fekken G, Veldman AEP, Iwanowski B, Buchner B. A volume-of-fluid based simulation method for wave impact problems. *Journal of Computational Physics* 2005; **206**:363–393.